الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جامعة وهران
Université d'Oran
كلية العلوم
Faculté des sciences

MEMOIRE

Présenté par

Mr SOLTANI
Réda

Pour obtenir

LE DIPLOME DE MAGISTER

Spécialité : Informatique
Option : Diagnostic, Aide à la Décision et Interaction Humain Machine

Intitulé :

Impact de la prise en compte d'information contextuelle sur la pertinence des systèmes de recommandations

Membres de jury :

M^{r}	G. BELALEM	Docteur, Université d'Oran, Algérie **(Président)**
M^{me}	S. NAIT-BAHLOUL	Docteur, Université d'Oran, Algérie **(Examinatrice)**
M^{r}	M. K. ABDI	Docteur, Université d'Oran, Algérie **(Examinateur)**
M^{me}	L. BABA-HAMED	Docteur, Université d'Oran, Algérie **(Encadreur)**

2011/2012

Dédicaces

Je dédie ce travail a ceux qui m'ont encouragé et soutenu

A ma mère et mon père qui ont partagé toutes mes joies et mes douleurs

A ma sœur Imène et mon frère Amine

A mes tantes Nadia, Kheira et Fouzia

A mes proches amis Sanaa, Nesrine, Karima, ainsi que Hichem, Mehdi, Boubakeur, Yazid, Lotfi, et Mohamed

A Mr SABRI Kamel et sa femme Amel

A tout le personnel de la conservation des forêts et la DCP

A toute la famille et tous mes amis et collègues que j'ai oublié de citer

Remerciements

Je remercie DIEU le tout puissant qui ma donné la force, la volonté et le courage pour accomplir ce travail.

Mes sincères remerciements vont à mon encadreur, madame Latifa BABA-HAMED, pour m'avoir dirigé et soutenu tout au long de ce projet. Votre esprit scientifique et votre souci pour de hautes performances sont une grande inspiration et m'ont poussé à me surpasser. Merci pour vos efforts, vos conseils et vos critiques constructives.

Je serai toujours reconnaissant envers mon co-encadreur Mr Sofiane ABBAR qui durant cette année n'a pas hésité à donner de son temps et à proposer son aide qui m'a été très précieuse.

Mes remerciements aux membres du jury qui ont accepté de juger mon travail.

Enfin j'exprime ma profonde reconnaissance à toutes les personnes qui ont contribué de près ou de loin au bon déroulement de ce travail.

"We are leaving the age of information and entering the age of recommendation"

Chris Anderson in The Long Tail

Résumé

Les systèmes de recommandation traditionnels apportent une solution au problème de surcharge d'informations. Ils fournissent aux utilisateurs les informations et contenus qui sont les plus pertinents pour eux. Ces systèmes ignorent le fait que des utilisateurs interagissent avec des systèmes selon un contexte particulier. A travers cette thèse, nous allons présenter un système de recommandation contextuel intrinsèque que nous pouvons appliquer pour la recommandation de contenus en général (livre, Url, article, produit, film, chanson, etc.). Nous illustrons l'approche adoptée pour affiner la recommandation de films à un utilisateur.

Mots-clés : *Système de recommandation, contexte, profil utilisateur, descriptif contenu, préférence, matching, contextualisation, similarité sémantique, similarité numérique, ontologie.*

Abstract

Traditional Recommender systems provide a solution to the problem of information overload. They provide to the users the information and content those are most relevant to them. These systems ignore the fact that users interact with systems in a particular context. Through this thesis, we present a recommendation system of intrinsic context that we can apply to the recommendation of general contents (book, Url, article, product, movie, song, etc). We illustrate the approach to refine the recommendation of movies to a user.

Keywords: *Recommendation system, context, user profile, content descriptive, preference, matching, contextualization, semantic similarity, digital similarity, ontology.*

Table des matières

LISTE DES FIGURES x
LISTE DES TABLEAUX xii
INTRODUCTION GENERALE 1

CHAPITRE I : Les Systèmes de Recommandation Classiques

1. Introduction 6
2. Définition 6
3. Ontlogies 7
    3.1 Définition 7
    3.2 Structure d'une ontologie 8
    3.3 Les différents types d'ontologie 8
4. Les principales approches de filtrage 9
    4.1 Filtrage collaboratif 9
        4.1.1 Définition 9
        4.1.2 Processus du filtrage collaboratif 10
        4.1.3 Profils et communautés 10
            4.1.3.1 Profil de l'utilisateur 11
            4.1.3.2 Communautés 11
        4.1.4 Exemple 11
    4.2 Filtrage basé sur le contenu 12
        4.2.1 Définition 12
        4.2.2 Descripteur d'article et profil utilisateur 13
        4.2.3Les différents types de matching 14
        4.2.4 Exemple 16
    4.3 Filtrage collaboratif vs approche basée sur le contenu 16
    4.4 Filtrage hybride 17
        4.4.1 Filtrage collaboratif via le contenu 17
        4.4.2 Filtrage collaboratif automatisé (ACF - Automated Collaborative Filtering) 19
        4.4.3 Exemple 19
5. Classification des approches de mesure de similarité 21
    5.1 Approches basées sur les arcs 21
        5.1.1 Mesure de Wu & Palmer (1994) 21
        5.1.2 Mesure de Rada et al (1989) 22
    5.2 Approches basées sur les nœuds 22
        5.2.1 Resnik (1999) 23
        5.2.2 Mesure de Lin (1998) 23
    5.3 Approches hybrides 23
        5.3.1 Mesure de Jiang et Conrath (1997) 23
        5.3.2 Mesure de Leacock et Chodorow (1998) 23
    5.4 Approches basées sur l'espace vectoriel 24
        5.4.1 Similarité de Cosine 24
        5.4.2 Similarité de Pearson 24

6. Limites des systèmes de recommandation 25
7. Les principaux travaux dans les systèmes de recommandation 26
8. Conclusion 28

CHAPITRE II : Les Systèmes de Recommandation Sensibles aux Contextes

1. Introduction 30
2. Définition de contexte 30
3. Importance du contexte dans les systèmes de recommandation 31
4. Dimensions du contexte 31
4.1. Contexte lié à l'utilisateur 31
4.2. Contexte lié à la requête 31
5. Comment obtenir le contexte ? 32
6. Pertinence de l'information contextuelle 32
7. Différentes méthodes d'introduction du contexte 32
7.1 L'approche de Pré-filtrage Contextuel (Pref) 32
7.2 L'approche de Post-filtrage Contextuel (PoF) 33
7.3 L'approche de la Modélisation Contextuelle (CM) 34
8. Les différents travaux sur les systèmes de recommandation sensibles au contexte 35
8.1 Exemples de SR sensibles au contexte 35
8.1.1 Micro-profiling 35
8.1.2 DaVI 36
8.1.3 DISCOVR 37
8.1.4New@hand 40
8.1.5 Extention de My Map 42
8.1.6 Sourcetone Interactive Radio - Musiques selon l'humeur 46
8.1.7 Amazon 46
8.1.8 Approche basée sur le Modèle d'Accès Personnalisé (PAM) 47
8.2 Comparaison 49
9. Conclusion 52

CHAPITRE III : Conception et Réalisation

1. Introduction 54
2. Modèle de données 54
3. Conception de l'application 57
3.1 Diagramme de classe 57
3.2 Diagramme de Séquence 57
3.3 Diagramme de collaboration 58
4. Architecture générale de l'application 59
5. Recommandation contextuelle 62
5.1 Création des profils utilisateurs 62
5.2 Résolution du contexte 63
5.3 Appariement profil/contenu 64
5.4. Mesures de similarités utilisées dans l'application 65
6. Exemple illustratif pour la recommandation des films 66
7. Conception de l'ontologie de l'application 68
7.1 Méthodologie de construction 68
7.2 Respect des principes de construction 68

7.3 Représentation hiérarchique de l'ontologie conceptuelle 69
8. Environnement de développement 70
8.1 Oracle 70
8.2 Langage de programmation java 70
8.2.1 NetBeans 71
8.3 Protégé 72
8.4 Jena 73
9. Conclusion 74

CHAPITRE IV: Expérimentation et Evaluation

1. Introduction 76
2. MovieLens et IMDB 76
3. Evaluation du système de recommandation 79
3.1 Construction des tables 79
3.2 Calcul de la prédiction 84
3.2.1 Similarité numérique 84
3.2.2 Similarité sémantique 84
3.2.3 Similarité globale 85
3.3 Recommandation contextuelle 85
3.4 La précision et le rappel 86
3.4.1 La précision 86
3.4.1.1La méthode des TOP(K) 86
3.4.1.2La méthode du seuil 87
3.4.2 Le rappel 88
3.5Interprétation de l'évaluation 89
3.5.1 Résultats de la méthode TOP(K) 89
3.5.2 Résultats de la méthode du seuil 95
4. Conclusion 96
CONCLUSION GENERALE 97
BIBLIOGRAPHIE 99

LISTE DES FIGURES

Figure 1. Schéma général du filtrage d'information 7
Figure 2. Classement des types d'ontologie selon leurs degrés d'abstraction 9
Figure 3. Principe général du filtrage collaboratif 12
Figure 4. Mesure de similarité hiérarchique 14
Figure 5. Exemple de recommandation faite par notebookreview.com 16
Figure 6. Hybridation classique de filtrage 20
Figure 7. Exemple d'un extrait d'ontologie 22
Figure 8. Exemple de contexte « période de l'année » 30
Figure 9. Pré-filtrage contextuel 33
Figure 10. Post-filtrage contextuel 33
Figure 11. Modélisation contextuelle 34
Figure 12. Types des composants de l'architecture DISCOVER. 38
Figure 13. Etape de traitement des services 39
Figure 14. Exemple d'interaction de plusieurs services 39
Figure 15. L'interface graphique du système News @ Hand 41
Figure 16. Expansion et contextualisation des préférences de l'utilisateur 42
Figure 17. Imprime écran du logiciel Inote 42
Figure 18. Fragment de métadonnée décrivant un restaurant 43
Figure 19. Un fragment MUP d'une personne nommé Dora 44
Figure 20. Exemple de règles étendant MyMap 45
Figure 21. Exemple de recommandation utilisant MyMap 45
Figure 22. Imprime écran du site www.sourcetone.com 46
Figure 23. Barre d'outils du site www.amazon.com 46
Figure 24. L'architecture du modèle d'accès personnalisé (PAM) 47
Figure 25. Architecture globale du système de recommandation sensible au contexte 49
Figure 26. Modèle Entité-Association 56
Figure 27. Diagramme de séquence 57
Figure 28. Diagramme de collaboration 58
Figure 29. Méthode d'introduction du contexte 59
Figure 30. Architecture générale de l'application 60
Figure 31. Représentation hiérarchique de l'Ontologie 69
Figure 32. Capture écran de l'éditeur NetBeans 72
Figure 33. Le site officiel du logiciel Protégé 72
Figure 34. La page d'accueil du site 73
Figure 35. Ajout de la librairie Jena 74
Figure 36. Les répartitions des évaluations 76
Figure 37. Table T-usernote 78
Figure 38. Table Training _set 78
Figure 39. Table Profil_user 78
Figure 40. Table Result_set 78
Figure 41. Méthode des TOPs k (Mesure de Jiang & Conrath Sans Contexte) 90
Figure 42. Méthode des TOPs k (Mesure de Pearson Sans Contexte) 90
Figure 43. Méthode des TOPs k (Mesure Globale Sans Contexte) 91
Figure 44. Méthode des TOPs k (Mesure de Jiang & Conrath Avec Contexte) 91
Figure 46. Méthode des TOPs k (Mesure globale Avec Contexte) 92
Figure 45. Méthode des TOPs k (Mesure de Pearson Avec Contexte) 92
Figure 47. Méthode des TOPs k (Mesure de Jiang & Conrath Avec et Sans Contexte pour TrainingeSet 70) 93

Figure 48. Méthode des TOPs k (Mesure de Pearson Avec et Sans Contexte pour TrainingSet 70)94
Figure 49. Méthode des TOPs k (Mesure Globale Avec et Sans Contexte pour TrainingeSet 70).. 94
Figure 50. Méthode du seuil (Mesure Globale Sans Contexte) 95
Figure 51. Mesure du seuil (Mesure Globale Avec Contexte)..... 96
Figure 52. Choix d'une session 104
Figure 53. Authentification de l'administrateur 105
Figure 54. Choix de la méthode de prédiction 105
Figure 55. Choix de la méthode de prédiction 106
Figure 56. Méthode des TOPs 106
Figure 57. Méthode de Seuil 107
Figure 58. Authentification de l'utilisateur 108
Figure 59. Liste des films pertinents 108

LISTE DES TABLEAUX

Tableau 1. Filtrage collaboratif basé sur les notes 18
Tableau 2. Filtrage collaboratif basé sur le contenu 18
Tableau 3. Filtrage collaboratif démographique 18
Tableau 4. Classification des recherches dans les systèmes de recommandation 27
Tableau 5. Matrice de similarité article-article 36
Tableau 6. Matrice de similarité avec l'attribut contextuel jour 37
Tableau 7. Tableau récapitulatif des différents Systèmes de recommandation sensibles aux contextes 51
Tableau 8. Profil de l'utilisateur N°424 66
Tableau 9. Matrice de similarités entre réalisateurs 67
Tableau 10. Enrichissement du profil utilisateur 67
Tableau 11. Dictionnaire de Base de Données 79
Tableau 12. Un extrait de la table T_usernote 79
Tableau 13. Un extrait de la table Training_set 80
Tableau 14. Un extrait de la table Result_set 81
Tableau 15. Un extrait de la table Classic_Profil_user 81
Tableau 16. Un extrait de la table Contextual_Profil_User 82
Tableau 17. Un extrait de la table Training_set pour le genre Action 82
Tableau 18. Un extrait de la table profil réalisateur 83
Tableau 19. Un extrait de la matrice réalisateur/réalisateur 83
Tableau 20. Illustrations du vecteur User pour le calcul de la similarité numérique 84
Tableau 21. Illustration du vecteur Film pour le calcul de la similarité numérique 84
Tableau 22. Illustrations d'une matrice pour le calcul de la similarité sémantique 84
Tableau 23. Résultat expérimental selon la similarité globale 86
Tableau 24. Exemple de la méthode des TOP(K) 86
Tableau 25. Exemple de la méthode du seuil 88
Tableau 26. Représentation des films avec leurs genres respectifs 110

INTRODUCTION GENERALE

La popularisation d'Internet ainsi que l'explosion des services de recommandation de nos jours ont propulsé la recherche d'information (RI) au premier plan. En effet, la surabondance de l'information a engendré la dégradation de la qualité des résultats retournés par un système de recommandation et a apporté des nouveaux problèmes au domaine de la RI.

Les systèmes de recommandations (SRs) sont des outils puissants aidant les utilisateurs, en mode en ligne, à résoudre le problème de la surcharge d'informations auquel ils sont confrontés aujourd'hui, avec l'avènement d'internet, en leur fournissant des recommandations personnalisées. Ce sont des systèmes de personnalisation qui présentent aux utilisateurs les contenus les plus pertinents, en utilisant certaines informations concernant leurs préférences passées. Les SRs utilisent des profils représentant des préférences relativement stables des utilisateurs pour calculer des recommandations. Ce calcul se fait par la prédiction des scores qu'un utilisateur est susceptible d'attribuer aux contenus. Les stratégies de recommandation se basent, généralement, sur le filtrage collaboratif (FC), le filtrage basé sur le contenu (FBC) ou sur une combinaison de ces deux approches. Le FC recommande des produits en se basant sur la similarité des préférences d'un groupe de clients connus comme étant des voisins. Cela suppose que les utilisateurs ayant des intérêts communs, dans le passé, continueront, probablement, à partager les mêmes goûts dans le futur. Les systèmes FBC utilisent l'appariement entre le profil d'un utilisateur et les descripteurs des contenus pour lui recommander les produits appropriés.

Les SRs traditionnels opèrent sur une matrice Utilisateur-Produit. Quand un utilisateur entre de nouveaux votes ou fait de nouveaux achats, son profile est mis à jour en ajoutant les nouvelles informations au vecteur de votes courant de cet utilisateur. Ces systèmes ignorent le fait que des utilisateurs interagissent avec des systèmes selon un contexte particulier, et que la note attribuée, par un utilisateur, à un produit, peut être différente en allant d'un contexte à un autre. En effet, l'endroit où se trouve l'individu (à la maison ou au travail), le temps (le matin ou le soir), la saison, l'état d'esprit peuvent influer sur les votes des utilisateurs, et ainsi améliorer l'exactitude des recommandations.

Netflix1 organise un challenge dans lequel l'entreprise offre 1 million de dollars pour celui ou celle qui réussit à implémenter un système de recommandations plus pertinent que le leur. Cette grosse somme d'argent montre le grand intérêt dont bénéficient les systèmes de recommandation auprès des industriels.

L'objectif de la présente thèse est de proposer une nouvelle approche permettant l'introduction du contexte dans le processus de la recommandation. Dans cette approche, nous définissons le contexte du point de vu des objets et non pas des utilisateurs. Le contexte est toute information qui influe sur l'intérêt que peut exprimer un utilisateur sur un objet. Nous distinguons deux types de contextes : contexte intrinsèque et contexte extrinsèque. Le premier type est défini sur l'ensemble des attributs relatif à l'objet lui-même. Par exemple, dans une relation Films le genre de film peut être considéré comme un attribut contextuel.
Le contexte extrinsèque est un ensemble d'attributs relatif à l'environnement d'interaction de l'utilisateur. Il peut renseigner sur l'emplacement géographique de l'utilisateur (adresse IP), sur la période temporelle de son interaction (date et heure), sur le media qu'il utilise pour l'interaction, etc. Très peu de travaux de recherche se sont intéressés au premier type de contexte. Dans le cadre de notre projet, nous avons utilisé le premier type de contexte (contexte faisant partie de l'objet lui-même) et avons montré, en le combinant avec le profil de l'utilisateur, comment il peut améliorer la précision de la recommandation et ainsi mieux répondre aux besoins des utilisateurs.

L'approche adoptée est générale dans le sens où nous pouvons l'appliquer pour la recommandation de contenus en général (livre, Url, article, produit, film, chanson, etc.). Nous illustrons son principe pour affiner la recommandation de films à un utilisateur.
L'évaluation d'un système utilisant cette nouvelle méthode a révélé de meilleurs résultats par rapport à ceux de l'évaluation d'un SR traditionnel (sans contexte).

Nos principales contributions sont les suivantes :

- La construction d'une ontologie de films, à travers laquelle nous montrons, essentiellement, les trois premières étapes de la construction d'une ontologie de domaine.
- Une nouvelle classification des mesures de similarité. Au cœur de la plupart des systèmes de recommandation, nous trouvons un opérateur de matching qui mesure la similarité de deux profils utilisateurs, de deux descripteurs de contenu ou bien la similarité entre un profil utilisateur et un descripteur de contenu. Nous avons donc été amenés à étudier les différentes mesures de similarités existantes et à les classer.
- L'élaboration et l'évaluation d'un système de recommandation sémantique sans contexte. Il s'agit ici d'un SR basé sur le FBC dont l'opérateur de matching utilise une combinaison linéaire d'une mesure de similarité numérique et d'une mesure de similarité sémantique. L'évaluation de ce système a donné des résultats satisfaisants en termes de précision (mesure bien connue dans le domaine de la recherche d'information).
- Une classification des travaux sur la contextualisation. Un état de l'art sur les différents SR sensibles au contexte a permis de dégager un certain nombre de critères sur lesquels nous nous sommes basés pour classifier ces systèmes.
- L'élaboration de deux algorithmes aidant à l'introduction du contexte. Le premier algorithme décrit le processus de création des profils à partir des logs transformés. L'idée principale de cet algorithme est de calculer le score contextuel de chaque prédicat dans chaque instance de contexte. Ceci permettra à notre système de capturer la variabilité des préférences en fonction des contextes. Le second décrit le processus de résolution des contextes. Ces deux algorithmes sont importants pour le développement de notre SR sensible au contexte.
- Evaluation du SR sensible au contexte et comparaison avec l'évaluation du SR sans contexte.

Notre mémoire est organisé comme suit :

Le premier chapitre est dédié aux systèmes de recommandation classiques (SRC). Nous définissons d'abord ce qu'est un système de recommandation. Ensuite nous détaillons ses différents types. Enfin, nous passons en revue les différentes mesures de similarité et présentons une classification de ces mesures.

Dans le deuxième chapitre, nous commençons par définir la notion de contexte. Nous enchaînons en montrant les différentes méthodes de son introduction dans le processus de recommandation. Nous terminons en présentant et comparant les différents SR sensibles au contexte.

Nous exposons notre approche dans le troisième chapitre. Nous commençons d'abord, par définir le modèle de données utilisé ainsi que la conception de notre application via les différents diagrammes UML. Ensuite nous expliquons la méthode adoptée pour l'introduction du contexte et nous détaillons l'architecture générale du SR adoptant notre nouvelle approche.

Nous y montrons, en détail, le fonctionnement des trois modules essentiels pour la résolution contextuelle, à savoir, *Création des profils utilisateurs*, *Résolution du contexte*, et l'*appariement profil/contenu*. Enfin, nous présentons les étapes suivies pour la construction de notre ontologie.

Le quatrième chapitre est consacré à l'évaluation du système et se compose de deux parties. La première consiste à l'évaluation du système sans tenir compte du contexte. La seconde partie concerne l'évaluation en tenant compte du contexte. L'illustration est faite dans le domaine cinématographique, le contexte intrinsèque choisi est « Réalisateurs ». Nous avons utilisé la mesure de similarité globale pour le calcul de la prédiction dans les deux parties ainsi que les métriques d'évaluation représentées par la Précision et le Rappel.

Nous clôturons ce mémoire par une conclusion et des perspectives. Enfin, nous présentons, en annexe 1, un manuel d'utilisation de notre application, et en annexe 2, un affichage des différents titres des films tirés des deux bases de données Movielens et IMDB .

CHAPITRE I

Les Systèmes de Recommandation Classiques

1. Introduction

Avec l'avènement d'internet, nous assistons aujourd'hui à une surcharge d'informations. Par exemple, une personne qui désire regarder un film se retrouve devant un volume très grand de propositions de films. Ce qui lui rend la tâche de choix d'un film très difficile. Les systèmes de recommandation sont apparus pour remédier à ce problème.

Dans ce chapitre, nous commençons d'abord par définir ce qu'est un système de recommandation ainsi que les notions qui s'y rapportent. Ensuite, nous présentons les trois approches principales de filtrage qui permettent la recommandation. Nous enchaînons en exposant les différentes mesures de similarité qui permettent, aux systèmes de recommandation, de faire l'appariement entre les concepts. Enfin, nous terminons en citant les limites et inconvénients des systèmes de recommandation ainsi que quelques principaux travaux dans le domaine.

2. Définition

Un système de filtrage d'information, ou système de recommandation (Recommender System) [30], est un filtre de flux entrant d'informations de façon personnalisée pour chaque utilisateur. Autrement dit, dans le but de personnaliser la recherche d'information dans un domaine d'application particulier, un système de filtrage collecte, sélectionne, classifie et suggère à l'utilisateur les informations qui répondent vraisemblablement à ses intérêts à long terme. Il existe à présent de nombreux systèmes de recommandation utilisés dans divers domaines comme la recherche documentaire, le commerce électronique, les loisirs, etc. On peut citer, à titre d'exemple, quelques sites Web populaires comme CiteSeer (pour la recommandation d'articles de recherche), Amazon (pour la recommandation des livres), MovieLens (pour la recommandation de films), LAST.FM (pour la recommandation de la musique) etc.

L'importance économique des systèmes de recommandation n'est plus à démontrer. L'entreprise Américaine de recommandation de film « Netflix », par exemple, organise un challenge dans lequel elle offre 1 million de dollar pour celui qui conçoit un SR plus performant que leur système.

Pour réaliser le filtrage, le système de recommandation (SR) utilise *les profils* représentant des préférences relativement stables des utilisateurs pour calculer des recommandations. Ce calcul se fait par la prédiction des scores qu'un utilisateur est susceptible d'attribuer aux contenus. Le SR adapte ce profil au cours du temps en exploitant au mieux le retour de pertinence que les utilisateurs fournissent sur les informations (documents) reçues. Par exemple, dans la figure 1, la fonction de décision du système traite le flux entrant de documents pour suggérer à l'utilisateur, en consultant son profil, les documents qu'il préfère. A son tour, l'utilisateur doit fournir ses évaluations, c'est-à-dire évaluer fréquemment les recommandations, pour que le système comprenne mieux ses besoins en information, et lui fournisse par conséquent de meilleures nouvelles recommandations.

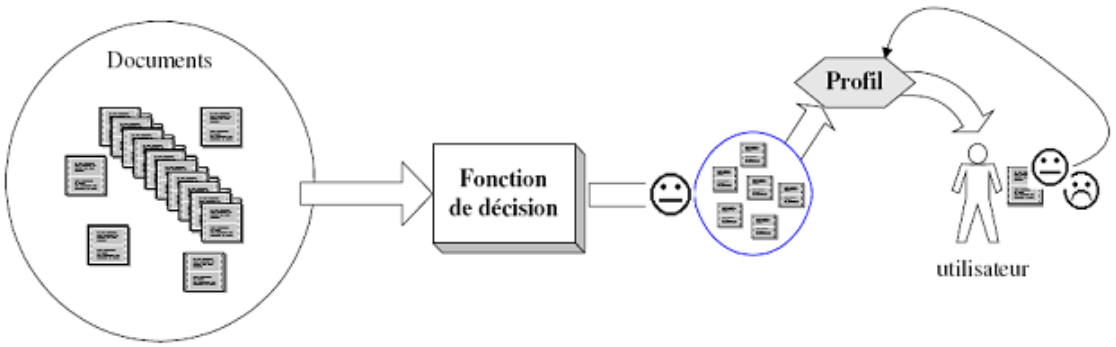

Figure 1. Schéma général du filtrage d'information

Les trois parties suivantes constituent un système de recommandation :

- **Les producteurs :** Ce sont ceux qui vont permettre de faire les recommandations, ils "fourniront" les données pour.
- **Le module de calcul :** Il s'agit de l'algorithme en lui même. En entrée il y a toutes les données et la demande et en sortie les différentes recommandations.
- **Le consommateur :** C'est celui qui demande la recommandation.

Comme nous l'avons déjà mentionné plus haut le calcul des recommandations se fait par la prédiction des scores qu'un utilisateur est susceptible d'attribuer aux contenus. Ce calcul nécessite souvent la présence d'une ontologie. Nous introduisons brièvement des notions relatives aux ontologies dans ce qui suit avant d'aborder les principales approches de filtrage.

3. Ontologies

Dans cette section, nous donnons la définition d'une ontologie, sa structure ainsi que ses différents types.

3.1 Définition

Le terme « ontologie », est construit à partir des racines grecques *ontos* (ce que existe, l'existant) et *logos* (le discours, l'étude). Le terme a été repris en informatique et en science de l'information au début des années 90, où une ontologie représente l'ensemble structuré des termes et concepts représentant le sens d'un champ d'informations, en effet, *Gruber* a défini l'ontologie comme *"La spécification d'une conceptualisation d'un domaine de connaissance"*. Plus simplement, elle permet de créer une base de données structurée dans laquelle il est possible de classifier des éléments par catégories et sous-catégories (concepts et sous-concepts), d'ajouter des définitions à chaque élément, de mettre des attributs et de créer tout type de relations (sémantiques ou taxonomiques (d'inclusion)) entre les éléments et les catégories.
En outre, et ce qui est loin d'être négligeable, les données stockées dans l'ontologie peuvent être facilement réutilisées par d'autres applications [18].

3.2 Structure d'une ontologie

Les ontologies sont représentées généralement par les éléments suivants [9] :

- **Concept :** Appelés aussi termes ou classes de l'ontologie, selon Gomez Pérez ces concepts peuvent être classifiés selon plusieurs dimensions :
    1. Niveau d'abstraction (concret ou abstrait).
    2. Atomicité (élémentaire ou composée).
    3. Niveau de réalité (réel ou fictif).

- **Instance** : C'est une définition extensionnelle de l'ontologie, par exemple les individus «Amina» et «Saloua» sont des instances du concept «personne».

- **Relation :** Les relations d'une ontologie désignent les différentes interactions et corrélations entre les concepts de l'ontologie, ces relations englobent les associations suivantes :

    1. Sous classe de (spécification ou généralisation),
    2. Partis de (agrégation ou composition),
    3. Associé a, instance de, est un ... etc.

- **Attributs** : Propriétés, fonctionnalités, caractéristiques ou paramètres que les objets peuvent posséder et partager.

3.3 Les différents types d'Ontologie

Les éléments de l'ontologie sont définis explicitement avec un langage dont la sémantique est plus ou moins formelle, dépendante du degré d'abstraction voulu. Par conséquent, les ontologies présentent des degrés d'abstraction différents selon le domaine auquel elles sont dédiées. On identifie trois types d'ontologies [19] selon un niveau d'abstraction décroissant: ***Ontologies d'applications, Ontologies de domaine***, ***Ontologies globales.***
Le schéma illustré dans la figure 2 résume les trois types d'ontologies présentées ci-dessus

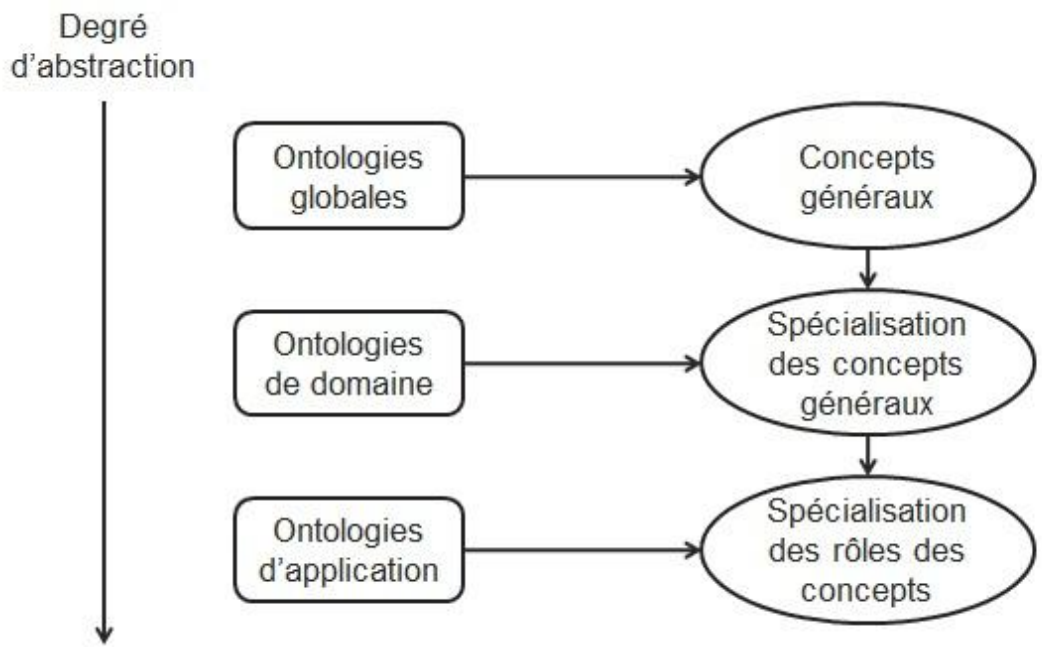

Figure 2. Classement des types d'ontologie selon leurs degrés d'abstraction

4. Les principales approches de filtrage

Il existe trois grandes approches de filtrage : basé sur le contenu, collaboratif et hybride [30]. Le filtrage basé sur le contenu compare les nouveaux documents au profil de l'utilisateur, et recommande ceux qui sont les plus proches. Le filtrage collaboratif compare les utilisateurs entre eux sur la base de leurs jugements passés pour créer des communautés, et chaque utilisateur reçoit les documents jugés pertinents par sa communauté. Le filtrage hybride combine le filtrage basé sur le contenu et le filtrage collaboratif pour exploiter au mieux les avantages de chacun.

4.1 Filtrage collaboratif

Dance cette section nous donnons dans un premier temps une définition du système de filtrage collaboratif. Ensuite, nous évoquons les différentes étapes par lesquelles passe le processus du filtrage collaboratif. Enfin, nous terminons par un exemple illustratif.

4.1.1 Définition

Le filtrage collaboratif (*Collaborative Filtering* « CF ») a pour principe d'exploiter les évaluations faites par des utilisateurs sur certains documents (contenus), afin de recommander ces mêmes documents à d'autres utilisateurs, et sans qu'il soit nécessaire d'analyser le contenu des documents.

Tous les utilisateurs du système de filtrage collaboratif peuvent tirer profit des évaluations des autres en recevant des recommandations pour lesquelles les utilisateurs les plus proches ont émis un jugement de valeur favorable, et cela sans que le système dispose d'un processus d'extraction du contenu des documents. Grâce à son indépendance vis-à-vis de la représentation des données, cette technique peut s'appliquer dans les contextes où le contenu est soit indisponible, soit difficile à analyser, et en particulier elle peut s'utiliser pour tout type de données : texte, image, audio et vidéo.

De plus, l'utilisateur est capable de découvrir divers domaines intéressants, car le principe du filtrage collaboratif ne se fonde absolument pas sur la dimension thématique des profils, et n'est pas soumis à l'effet « entonnoir ».
Un autre avantage du filtrage collaboratif est que les jugements de valeur des utilisateurs intègrent non seulement la dimension thématique mais aussi d'autres facteurs relatifs à la qualité des documents tels que la diversité, la nouveauté, l'adéquation du public visé, etc.

Le CF souffre de plusieurs gros problèmes. Le problème principal étant le démarrage à froid : c'est le fait qu'un utilisateur doit voter sur beaucoup (trop) d'objets avant d'obtenir des recommandations.

4.1.2 Processus du filtrage collaboratif

Le processus du filtrage collaboratif suit les étapes données ci-dessous :

4.1.2.1 Evaluation des recommandations

Selon le principe de base du filtrage collaboratif, les utilisateurs doivent fournir leurs évaluations sur des documents afin que le système forme les communautés. Evaluer une recommandation peut se faire de façon explicite ou implicite, comme suit :
– **Explicite** : L'utilisateur donne une valeur numérique sur une échelle donnée (par exemple, de 1 à 5, ou de 1 à 10, etc.), ou bien, une valeur qualitative de satisfaction, par exemple, *mauvaise*, *moyenne*, *bonne* et *excellente*.
– **Implicite** : Le système induit la satisfaction de l'utilisateur à travers ses actions. Par exemple, le système estimera qu'une recommandation supprimée correspond à une évaluation très mauvaise, alors qu'une recommandation imprimée ou sauvegardée peut être interprétée comme une bonne évaluation. Il faut par ailleurs noter que les recommandations qu'évalue un utilisateur peuvent être générées par le système et/ou choisies par l'utilisateur lui-même.

4.1.2.2 Formation des communautés

Le processus de formation des communautés est le noyau d'un système de filtrage collaboratif.
Pour chaque utilisateur, le système doit calculer sa communauté, généralement cela se fait par la proximité des évaluations des utilisateurs. Pour ce faire, on peut calculer, dans un premier temps, la proximité entre un utilisateur donné et tous les autres. Ensuite, et afin de créer concrètement la communauté de l'utilisateur, on applique souvent la méthode des voisins les plus proches en utilisant un seuil pour le niveau de proximité ou un seuil pour la taille maximale de la communauté, en raison de sa performance et sa précision.

4.1.2.3 Production des recommandations

Dans ce dernier processus, une fois la communauté de l'utilisateur créée, le système prédit l'intérêt qu'un document particulier peut présenter pour l'utilisateur en s'appuyant sur les évaluations que les membres de la communauté ont faites sur ce même document. Lorsque l'intérêt prédit dépasse un certain seuil, le système recommande le document à l'utilisateur.

4.1.3 Profils et communautés

Ici, nous discutons les profils basés sur l'historique des évaluations des utilisateurs, ainsi que les communautés, qui sont les deux facteurs clés d'un système de filtrage collaboratif.
Le problème de la surcharge d'information peut être pallié par la personnalisation de l'accès aux informations, en utilisant des *profils* représentant des intérêts relativement stables des utilisateurs. En d'autres termes, les profils des utilisateurs sont utilisés comme des critères

persistants dans la recherche d'information. Le filtrage d'information (*Information Filtering*) s'appuie également sur de tels profils pour filtrer un flux de documents en vue de ne conserver que les documents pertinents pour l'utilisateur.

4.1.3.1 Profil de l'utilisateur

Le profil de l'utilisateur est composé d'un ensemble de prédicats pondérés. Le poids d'un prédicat exprime son intérêt relatif pour l'utilisateur. Il est spécifié par un nombre réel compris entre 0 et 1. Le profil s'enrichit progressivement au fur et à mesure que l'utilisateur évalue des documents reçus.

Outre les informations d'identification de base (par exemple, l'identifiant ou des éléments d'état civil), le profil de l'utilisateur peut regrouper des informations très diverses selon les besoins.
Parmi celles-ci, nous pouvons citer :

- Des caractéristiques personnelles pouvant influencer fortement l'interaction (âge, sexe, etc).
- Les intérêts et les préférences générales de l'utilisateur relatives à la tâche à accomplir, qui permettent une adaptation à ses attentes.
- Qualité. Cette dimension contient tous les facteurs reflétant les préférences relatives à la qualité de l'information, comme la disponibilité de données, la concision, le style et la structure du document, etc. Dans cette dimension, nous nous intéressons en particulier à la diversité de l'information.
- Sécurité. La dimension de sécurité dans le contexte du filtrage collaboratif, est le niveau de confidentialité concernant tous les autres critères.
- Livraison. On peut citer à titre d'exemple la modalité de livraison des informations, comme le format, le standard, le volume et le mode de visualisation, le délai et le prix, etc.
- Un historique des interactions avec le service, qui peuvent permettre de modéliser les habitudes comportementales.

4.1.3.2 Communautés

La notion de communauté dans un système de filtrage collaboratif est définie comme le regroupement des utilisateurs en fonction de l'historique de leurs évaluations, afin que le système calcule des recommandations. Selon cette optique, les profils sont un facteur interactif, alors que les communautés sont considérées comme un facteur interne du système.

4.1.4 Exemple

Dans la figure 3, nous schématisons le principe du filtrage collaboratif. On suppose que l'on a des communautés formées par la proximité des évaluations des utilisateurs. Le document *d* sera recommandé à l'utilisateur *u*, car ce document est apprécié par la communauté G où se trouve l'utilisateur.

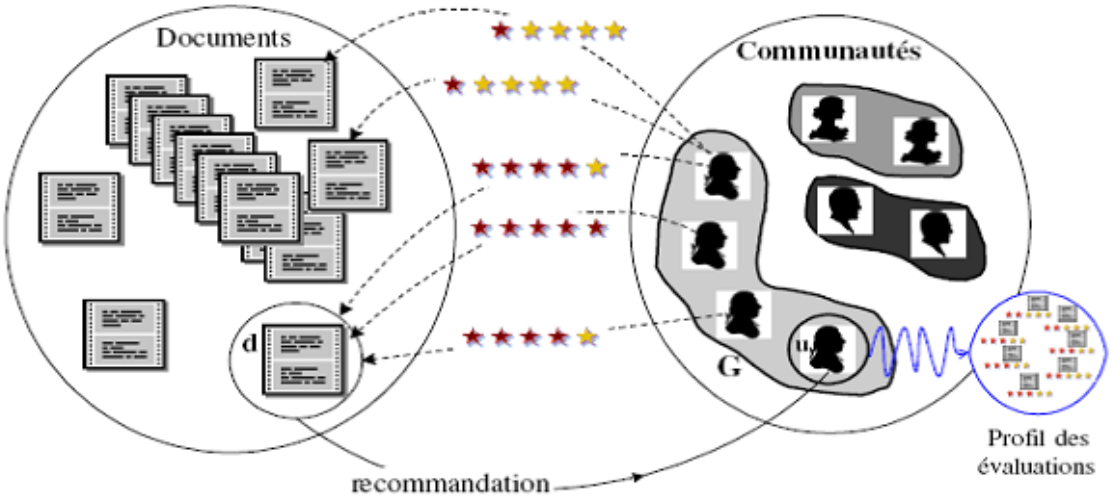

Figure 3. Principe général du filtrage collaboratif

4.2 Filtrage basé sur le contenu

Dans cette section, nous définissons dans un premier temps, ce qu'est le filtrage basé sur le contenu. Ensuite, nous présentons les différents types de matching qui existent. Enfin, nous donnons un exemple pour illustrer le fonctionnement d'un système utilisant le filtrage basé sur le contenu.

4.2.1 Définition

Le filtrage basé sur le contenu (*Content-based Filtering*) [30], qui est une évolution générale des études sur le filtrage d'information, s'appuie sur le contenu des documents (thèmes abordés) pour les comparer à un profil lui-même constitué de thèmes. Chaque utilisateur du système possède alors un profil qui décrit ses propres centres d'intérêt. Par exemple, le profil peut contenir une liste des thèmes ou préférences que l'utilisateur aime bien ou qu'il n'aime pas. Lors de l'arrivée d'un nouveau document, le système compare le descriptif du document avec le profil de l'utilisateur pour prédire l'utilité de ce document pour cet utilisateur.
Le premier avantage du filtrage basé sur le contenu est qu'il peut répondre aux préférences de l'utilisateur à long terme, en employant des techniques efficaces dans le domaine de l'intelligence artificielle pour la mise à jour des profils et le recoupement entre profils et documents. En outre, l'utilisateur dans un tel système ne dépend absolument pas des autres. Ainsi, il recevra des recommandations du système même s'il est le seul inscrit, pour peu qu'il ait décrit son profil en donnant un ensemble de thèmes qui l'intéressent.
En revanche, cette technique de filtrage est soumise à l'effet « entonnoir », car le profil de l'utilisateur évolue naturellement par restriction progressive sur les thèmes recherchés. Ainsi, l'utilisateur ne reçoit que les recommandations relatives aux thèmes présentés dans son profil, une fois devenu stable. Par conséquent, il ne peut pas découvrir de nouveaux domaines potentiellement intéressants pour lui. Le filtrage basé sur le contenu est également victime d'un effet de masse, puisqu'il ne bénéficie pas des jugements de qualité des autres utilisateurs.

4.2.2 Descripteur d'article et profil utilisateur

L'algorithme de filtrage basé sur le contenu peut réaliser le matching entre un descripteur de contenu (comme par exemple, documents, News, Url, livre, etc.) et un profile utilisateur et détermine le degré de pertinence de chaque article (ou contenu) pour les utilisateurs potentiels. Si de nombreux articles s'accumulent dans un certain laps de temps, l'algorithme de filtrage de contenu peut ordonner les articles en fonction de leur pertinence pour chacun des utilisateurs potentiels [28].

4.2.2.1 Représentation des contenus - le descripteur d'article

Un descripteur d'article se compose d'un ensemble de concepts qui peuvent être représentés par une ontologie de domaine. Les concepts qui représentent un élément sont les plus spécialisés dans une branche de la hiérarchie. Par exemple, si un article traite le « sport » et en particulier le « football », il est représenté par «football» seulement.
De toute évidence, un article peut être représenté avec de nombreux concepts de l'ontologie, chaque concept peut apparaître dans n'importe quelle branche de la hiérarchie de l'ontologie et à tout niveau – cela dépend du contenu réel de cet article. Il est à noter que le profil peut inclure des concepts frères, c'est-à-dire les fils d'un même concept. Par exemple, un profil d'article peut inclure à la fois « football » et « basketball » les fils de « sport ».

4.2.2.2 Représentation des utilisateurs - le profil d'utilisateur

Un profil d'utilisateur basé sur le contenu se compose d'une liste pondérée de concepts de l'ontologie, représentant ses préférences (ses intérêts).
De toute évidence, le profil de l'utilisateur peut comporter de nombreux concepts de l'ontologie, chacun figurant dans les différentes branches et différents niveaux de la hiérarchie. Par exemple, le profil de l'utilisateur peut inclure uniquement « sport », ou « sport » et « football », ou « football » et « basketball », ou tous les trois - en plus de nombreux autres concepts. Cela signifie qu'un certain concept dans un descripteur d'article peut être « comparé », avec plus d'un concept équivalent dans le profil de l'utilisateur. Par exemple, si un descripteur d'article inclut «football» et le profil de l'utilisateur inclut à la fois « sport » et « football », alors il est de type « Matching parfait » entre les deux profils en raison du concept commun « football », et aussi « Matching partiel » à cause du concept père « sport ».

Comme indiqué précédemment, le profil de l'utilisateur basé sur le contenu peut être initialement généré par l'utilisateur qui sélectionne les concepts depuis l'ontologie et de leur attribuer des poids d'importance. (Le total des poids est normalisé à 100%). Ensuite, le profil de l'utilisateur est mis à jour au fil du le temps en fonction de la réaction implicite de l'utilisateur : lorsque l'utilisateur lit un article et le trouve intéressant, les concepts dans le descripteur de cet article qui n'étaient pas initialement dans le profil utilisateur, sont ajoutés à celui-ci, et les poids de tous les concepts dans ce profil sont recalculés en fonction du nombre de clicks.

4.2.2.3 Similarités entre un descripteur d'article et un profil utilisateur

Un descripteur d'article et un profil utilisateur sont semblables à un certain degré si leurs profils comprennent des concepts communs (le même) ou des concepts relatifs, c'est-à-dire des concepts ayant une sorte de relation père-fils. Un descripteur d'article et un profil utilisateur peuvent avoir de nombreux concepts communs ou relatifs; de toute évidence, plus les concepts sont communs ou relatifs, plus forte est leur similitude. Par exemple, si le profil de l'utilisateur inclut « football » et « sport », ce profil est similaire (à un certain degré) à un article qui comprend ces deux concepts, mais il est moins semblable à un article incluant juste

« sport », et il est plus semblable à un article, comprenant « sport » et « football » et « basketball ».

4.2.3Les différents types de matching

Dans une ontologie hiérarchique à 3 niveaux, nous pouvons distinguer entre 9 différents cas de similitude entre les concepts dans un profil utilisateur et un descripteur d'article [28], comme représenté sur la figure 4.

Dans les paragraphes qui suivent, nous présentons les différents types du matching qui existent.

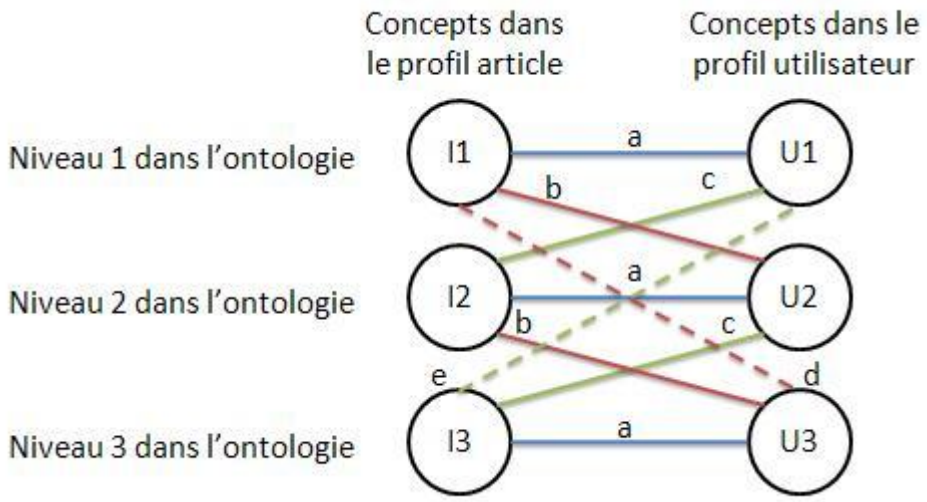

Figure 4. Mesure de similarité hiérarchique

Matching parfait

Le concept apparait à la fois dans le profil de l'utilisateur et le profil d'article. I1, I2, I3 indiquent le niveau d'un concept dans un profil article, et U1, U2, U3 le niveau d'un concept dans le profil utilisateur.

Un «Matching parfait» peut survenir dans 3 cas:

- I1 = U1 (par exemple, les deux profils articles et utilisateurs incluent « sport »).
- I2 = U2 (par exemple, les deux profils articles et utilisateurs incluent « football »).
- I3 = U3 (par exemple, les deux profils articles et utilisateurs incluent « jeu mondial »).

Matching partiel

Un concept apparaît seulement dans un des profils, alors que le père ou le fils de ce concept apparaît dans l'autre profil. Un « Matching partiel » peut se produire dans les 2 paires de cas:

- I1 = U2 (par exemple un concept article est « sport », tandis que le concept utilisateur est « football »).
- I2 = U3 (par exemple un concept article est « football », tandis que le concept utilisateur est « jeu mondial »).

Dans ces 2 cas, le concept article est plus général que le concept utilisateur (un seul niveau de différence), c'est-à-dire l'intérêt de l'utilisateur est plus précis / spécifique que l'article.

- I2 = U1 (par exemple un concept article est « basketball », tandis que le concept utilisateur est « sport »).
- I3 = U2 (par exemple un concept article est « ligue d'Euro », tandis que le concept utilisateur est « basketball »).

Dans ces 2 cas, le concept article est plus spécifique que le concept utilisateur, c'est-à-dire l'intérêt de l'utilisateur est plus général.

Matching faible

Un concept apparaît dans un profil, alors qu'un concept grand-père ou concept petit-fils apparaît dans l'autre profil (les concepts sont séparés pas 2 niveaux). Un matching faible peut se produire dans 2 cas:

- I1 = U3 (par exemple un concept article est « sport », tandis que le concept utilisateur est « jeu mondial ») dans ce cas, l'article est beaucoup plus général que l'intérêt de l'utilisateur.
- I3 = U1 (par exemple un concept article est « ligue d'Euro », tandis que le concept utilisateur est « sport ») dans ce cas, l'article est beaucoup plus précis que l'intérêt de l'utilisateur.

Pour chacun de ces 9 cas possibles nous déterminons un score de similarité. Dans les 3 cas de « Matching parfait » marqué « a » (voir Figure 4), le score est de 1 (maximum), dans tous les autres cas, le score doit être inférieur à 1, cela dépend s'il s'agit d'un matching « partiel » ou « faible » et de la « direction » de la relation, c'est-à-dire, si le concept utilisateur est plus général ou plus spécifique que le concept article. Par exemple, le score pour le cas I1 = U2 (le concept article est plus général que le concept utilisateur), exemple 2/5, alors que le cas I2 = U1 (le concept article est plus spécifique que le concept utilisateur) peut être (supérieur) par exemple 2/3. La raison de cela peut être que dans le premier cas, l'article traite avec un concept plus général que l'intérêt de l'utilisateur, rapportant une précision inférieur que dans les autres cas, ou les articles traitent un concept plus spécifique que l'intérêt de l'utilisateur, donc rapporte une précision supérieur. Mais cette hypothèse, ainsi que le score exact de la similitude de tous les cas est soumise à l'expérimentation.

Nous donnons dans ce qui suit un arrangement de marquage possible pour les 9 cas possibles de la figure 4:

- I1 = U1 → 1 | I2 = U2 → 1 | I3 = U3 →1 (3 cas de « Matching parfait », marqué 'a' dans la figure 4).
- I1 = U2 → 2/5 | I2 = U3 → 2/5 (2 cas de « Matching partiel » le concept article est plus général, marqué 'b').
- I2 = U1 → 2/3 | I3 = U2 → 2/3 (2 cas de « Matching partiel » le concept article est plus spécifique, marqué 'c').
- I1 = U3 → 1/3 (cas de « Matching faible » le concept article est beaucoup plus général, marqué 'd').
- I3 = U1 → 1/2 (cas de « Matching faible » le concept article est beaucoup plus précis, marqué 'e').

4.2.4 Exemple

La figure 5 montre un exemple d'un système de recommandation utilisant le filtrage basé sur le contenu. Nous remarquons sur la partie droite de cette figure, les choix de l'utilisateur concernant l'achat d'un PC portable. La partie gauche de la figure présente les trois résultats proposés par le système.

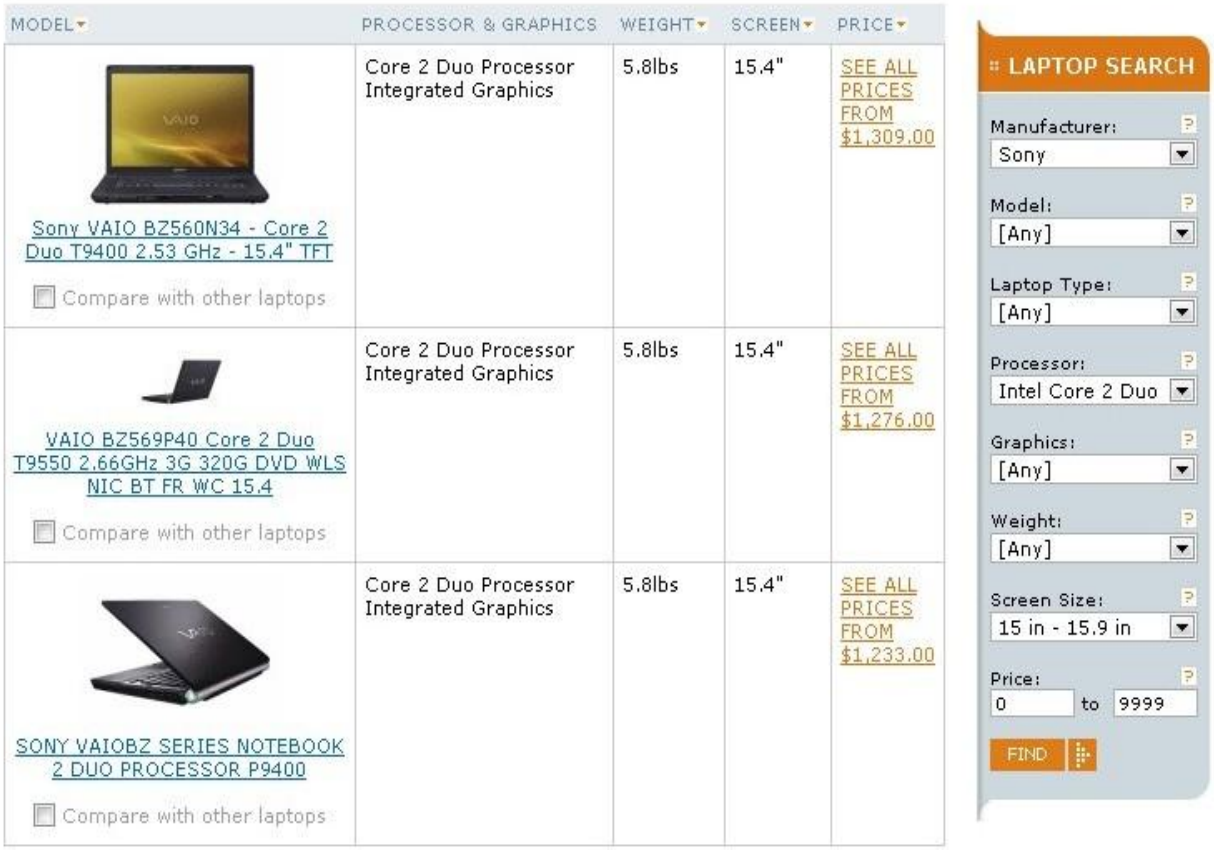

Figure 5. Exemple de recommandation faite par notebookreview.com

4.3 Filtrage collaboratif vs approche basée sur le contenu

À la base, nous avions des informations sur le parcours hypertextuel de nos clients. Dans un premier temps, nous en avons déduit quelles sont les pages qui ont plu, quelles sont celles qui ont déplu, et quelles sont celles qui n'ont pas été visitées. Ceci nous a permis de former un premier profil pour nos clients. Pour cela, à chaque utilisateur nous associons à un vecteur de notes sur les pages qu'il a parcourues. C'est cette représentation de profil qui est utilisée pour le filtrage collaboratif.

Par contre, la méthode d'analyse de contenu ne s'arrête pas là. Pour chaque page identifiée, une analyse en profondeur du texte va permettre d'en extraire les caractéristiques essentielles, pour tenter de mieux comprendre ce qui plait ou déplait dans la page au client, et former un autre profil en associant en plus à chaque utilisateur un vecteur des thèmes (et éventuellement des mots) les plus fréquemment rencontrés dans ses lectures, et constituer ainsi une mémoire des centres d'intérêts du client. Chacune des approches possède ses limitations. Celles de l'analyse basée sur le contenu sont les suivantes:

- Avant d'analyser le contenu, les articles doivent, dans un premier temps, être mis sous une forme convenable pour l'algorithme. Cependant, des nouvelles technologies telles le son, les photos, ou la vidéo, etc. ne peuvent être analysées automatiquement pour obtenir des informations suffisamment pertinentes.

- Etant donné que relativement peu de textes ont des vecteurs vraiment similaires, le système recommande souvent ce que l'utilisateur a déjà vu auparavant.

- À part si le système utilise un niveau d'analyse linguistique très poussé, il ne peut filtrer des articles en se basant sur leur qualité, style ou point de vue. Par exemple, si on se restreint à une approche bag-of-words, et si deux articles contiennent les mêmes termes, on ne pourra distinguer entre un bien écrit et un autre mal écrit.

Au contraire, le filtrage collaboratif n'utilisant aucune information concernant le contenu des articles (texte, image, musique, etc.), les articles filtrés n'ont pas besoin d'être préalablement analysés grammaticalement. De plus, ce système étant basé sur l'évaluation des objets par les utilisateurs, les articles recommandés peuvent alors être très différents de ceux que l'utilisateur a déjà vus auparavant, et les recommandations sont basées sur une évaluation implicite de la qualité des articles, perçue par les utilisateurs. Mais par contre, trois limites sont présentes pour cette approche:

- La première est que la recommandation à un client dépend des autres clients.

- La seconde est qu'il faut associer à chaque client la liste des textes lus.

- Et la troisième est le fait que le système repose sur l'hypothèse implicite que toutes les caractéristiques d'un document sont de la même importance pour l'évaluation de ce document par un utilisateur. Or, deux utilisateurs peuvent donner tous deux une bonne (ou inversement une mauvaise) note à un même document pour des raisons complètement différentes.

4.4 Filtrage hybride

La recommandation peut être faite lorsqu'un article est bien apprécié par un ensemble d'utilisateurs qui présentent des profils similaires ou par l'utilisateur lui-même.
L'idée principale de l'approche par hybridation est de proposer au nouvel utilisateur d'adopter un profil prédéfini selon le cas, afin de recevoir des recommandations exploratoires [11].
En général, l'hybridation dans un système de filtrage se réalise en deux phases : *préliminaire* et *couplage*.

4.4.1 Filtrage collaboratif via le contenu

Le filtrage collaboratif tel que nous l'avons présenté précédemment utilise l'ensemble des votes des utilisateurs sur les articles pour leur attribuer un degré de similarité. La matrice de données utilisée ressemble alors au tableau 1 suivant:

Tableau 1. Filtrage collaboratif basé sur les notes

	Article 1	Article 2	Article 3	Article 4	Article 5 considéré
Utilisateur 1	-	-	7	6	???
Utilisateur 2	-	-	5	6	1
Utilisateur 3	-	-	6	6	3
Utilisateur 4	7	5	-	-	6
Utilisateur 5	7	6	-	-	4

Ici, on cherche la note que l'utilisateur 1 va attribuer à l'article 5, en fonction des notes que les autres utilisateurs ont attribué à l'article, et de la similarité entre leurs vecteurs de notes sur les autres articles.

L'idée du *filtrage collaboratif via le contenu* est d'utiliser les profils thématiques des clients, formés à partir de l'analyse du contenu des articles qu'ils ont lus, pour comparer les utilisateurs dans la phase de calcul des corrélations entre utilisateurs du filtrage collaboratif. La matrice de données utilisée ressemble alors au tableau 2 suivant:

Tableau 2. Filtrage collaboratif basé sur le contenu

	Mot 1	Mot 2	Mot 3	Mot 4	Mot 5	Article 5 considéré
Utilisateur 1	2.5	0	0.2	0	0	???
Utilisateur 2	1.1	0	1.1	1.5	0	1
Utilisateur 3	1.5	0	3.5	1.5	0.5	3
Utilisateur 4	1.1	1.1	2.1	2.0	2.5	6
Utilisateur 5	1.1	2.2	0	0	3.5	4

Cette fois, on cherche la note que l'utilisateur 1 va attribuer à l'article 5, en fonction des notes que les autres utilisateurs ont attribué à l'article, et de la similarité entre leurs vecteurs de mots, représentant l'ensemble des thèmes identifiés comme préférés (récurrents) dans leurs lectures.
Une autre alternative possible, si nous possédons des informations personnelles sur les clients, est le *filtrage collaboratif démographique*. Celui-ci consiste à utiliser des informations personnelles sur les clients, du type âge, sexe, Catégorie Socioprofessionnelle, lieu d'habitation, etc., pour les comparer dans la phase de calcul des corrélations entre utilisateurs du filtrage collaboratif. La matrice de données utilisée ressemble alors au tableau 3 suivant:

Tableau 3. Filtrage collaboratif démographique

	Age	Sexe	CSP	Code habitation	Article 5 considéré
Utilisateur 1	22	M	1	59	???
Utilisateur 2	35	M	2	62	1
Utilisateur 3	18	F	1	59	3
Utilisateur 4	87	F	3	95	6
Utilisateur 5	29	M	1	62	4

4.4.2 Filtrage collaboratif automatisé (*ACF - Automated Collaborative Filtering*)

L'idée qui a été développée ici, est de combiner l'utilisation des caractéristiques essentielles des documents avec les évaluations des documents par les clients, c'est à dire de combiner les deux méthodes, qui paraissent complémentaires, de filtrage collaboratif et d'analyse de contenu.

Ceci permettrait de réduire le nombre d'articles à considérer pour l'algorithme de prédiction, et permettrait également d'obtenir de meilleures prédictions, puisque seuls les articles ayant des caractéristiques proches de celles du client considéré seront pris en compte pour calculer les prédictions.

Le système consiste donc en deux composantes principales: un *agent personnel d'information* pour chaque utilisateur contrôle continuellement les intérêts de l'utilisateur, recommande des documents potentiellement intéressants, et collecte et propage l'évaluation des documents par l'utilisateur; et un *serveur de filtrage collaboratif automatique basé sur les caractéristiques (FGACF-Feature-Guided Automated Collaborative Filtering)* collecte les évaluations de plusieurs agents utilisateurs pour recommander ensuite de nouveaux documents à ces utilisateurs.

En fait, les agents utilisateurs aident à structurer l'ensemble des documents favoris d'un utilisateur. Ils apprennent les intérêts de l'utilisateur, collectent les évaluations des documents, effectuent des recommandations de documents, et peuvent aider à localiser un type particulier de documents.

Le serveur FGACF, lui, s'occupe de collecter les évaluations des agents utilisateurs. De plus, il contient un module de traitement des documents qui permet d'extraire des caractéristiques simples d'un document, qui seront utilisées pour déterminer une partition de l'espace des documents.

L'algorithme ACF peut être guidé par les caractéristiques des documents de deux façons: soit des clusters sont automatiquement formés à l'aide des corrélations entre utilisateurs, puis analysés pour trouver leurs points communs; soit on utilise les caractéristiques du document pour déterminer une partition de l'espace des documents

4.4.3 Exemple

Comme illustré dans la figure 6. D'abord, le système applique à part une ou plusieurs techniques de filtrage de base telles que filtrage collaboratif, filtrage basé sur le contenu et filtrage démographique. Cette phase préliminaire génère les recommandations candidates, et qui sont combinées par certaines méthodes d'hybridation dans la phase de couplage, afin de produire les recommandations finales destinées à l'utilisateur.

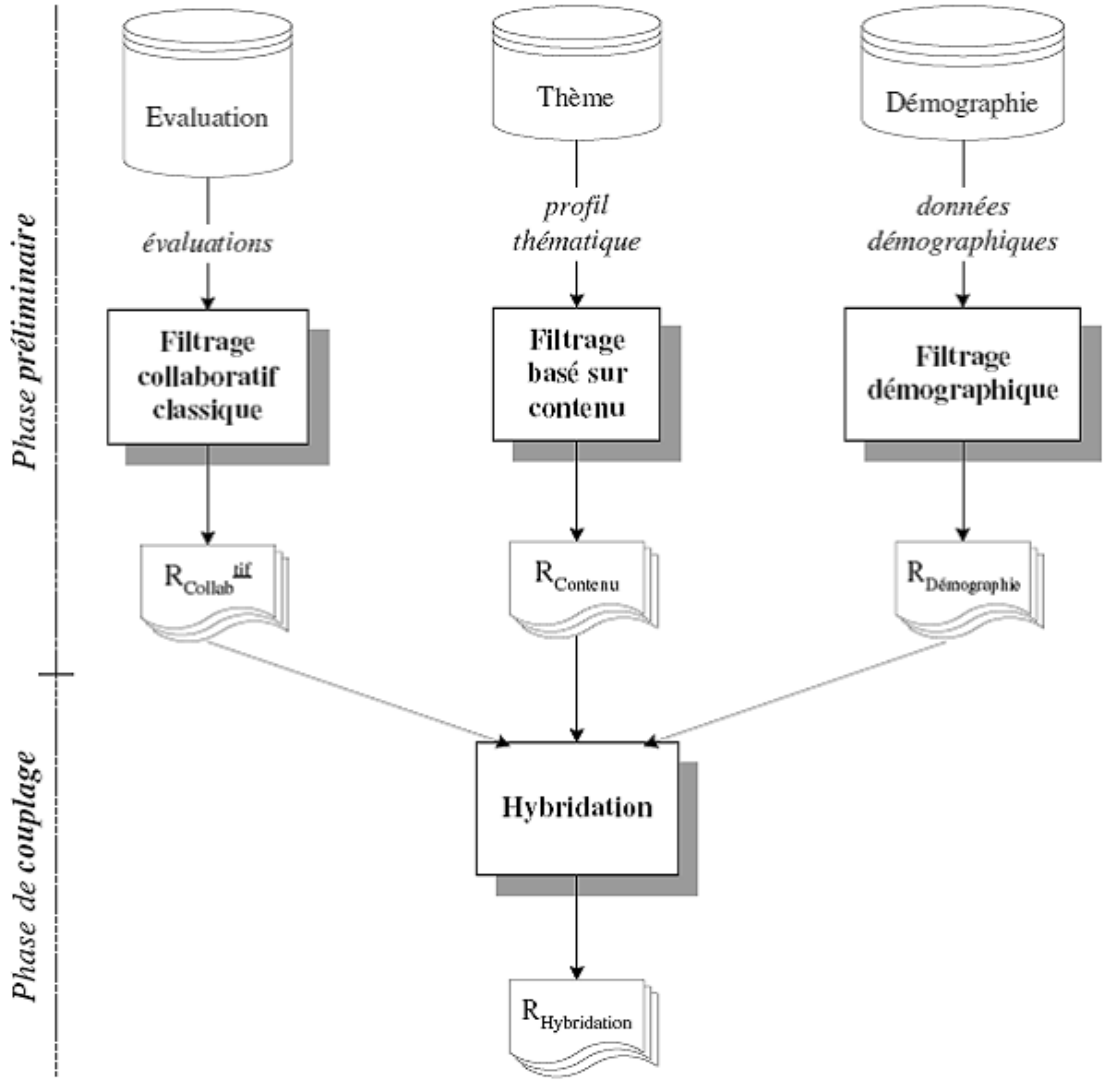

Figure 6. Hybridation classique de filtrage

Comme nous l'avons déjà mentionné précédemment, au cœur de la plupart des systèmes de recommandation, nous trouvons un opérateur de matching qui mesure la similarité de deux profils utilisateurs, de deux descripteurs de contenu ou bien la similarité entre un profil utilisateur et un descripteur de contenu. Comme les profils utilisateurs et les descripteurs de contenu sont souvent modélisés avec des vecteurs de mots clés pondérés, seules les mesures vectorielles comme Cosinus et corrélation de Pearson sont utilisées. Or, l'avènement du web sémantique et le développement des ontologies ont mis à notre disposition une panoplie de mesures de similarité sémantiques qui peuvent compléter les mesures vectorielles citées ci-dessus. Pour montrer les possibilités offertes par les mesures de similarité sémantiques, considérons l'exemple suivant.

Exemple :
Soit un utilisateur, Bill, qui renseigne trois de ses préférences cinématographiques ProfilBill{(Action, 5), (Guerre, 4), (Thriller, 5)}. Soient les deux films suivants avec leurs descriptions (limitées aux genres) : Film1{Aventure, Policier}, Film2{Drame, Comédie}. Les systèmes de recommandation classiques, basés sur les similarités vectorielles seront incapables de dire lequel de ces deux films il faut recommander à Bill. Or, grâce aux ontologies, nous savons que les concepts Aventure et Policier caractérisant Film1 sont plus proches (similaires) des concepts Action, Guerre et Thriller du profil de Bill que ne le sont les concepts Drame et Comédie du Film2. Ainsi, si une recommandation est demandée, c'est le film Film1 qui doit être envoyé à Bill. On voit à travers cet exemple que l'aspect sémantique suffit à lui seul, dans certains cas, pour ordonner des contenus par rapport à un profil utilisateur en fonction de leurs similarités.

Nous avons jugé important de présenter quelques mesures de similarité des plus connues. La section suivante montre une classification de ces mesures.

5. Classification des approches de mesure de similarité

Dans cette classification, nous distinguons quatre grandes catégories de mesures de similarité [35]. Nous avons proposé une autre classification dans [6].

5.1 Approches basées sur les arcs

La mesure de similarité la plus intuitive des objets dans une ontologie est leurs distances. Cette similarité est évaluée par la distance qui sépare les objets dans l'ontologie.
Ces mesures se servent de la structure hiérarchique de l'ontologie pour déterminer la similarité sémantique entre les concepts. Le calcul des distances dans l'ontologie est basé sur un graphe de spécialisation des objets. Dans chaque graphe, la distance de l'ontologie doit être caractérisée par le plus court chemin qui fait intervenir un ancêtre commun ou le plus petit généralisant, connectant potentiellement deux objets à travers des descendants communs. Parmi les travaux classifiés sous cette catégorie on peut citer :

5.1.1 Mesure de Wu & Palmer (1994)

La mesure de similarité de Wu et Palmer [39] est basée sur le principe suivant : Etant donnée une ontologie formée par un ensemble de nœuds et un nœud racine R (Figure 7). Soit X et Y deux éléments de l'ontologie dont nous allons calculer la similarité. Le principe de calcul de similarité est basé sur les distances (N1 et N2) qui séparent les nœuds X et Y du nœud racine et la distance qui sépare le concept subsumant (CS) de X et de Y du nœud R.

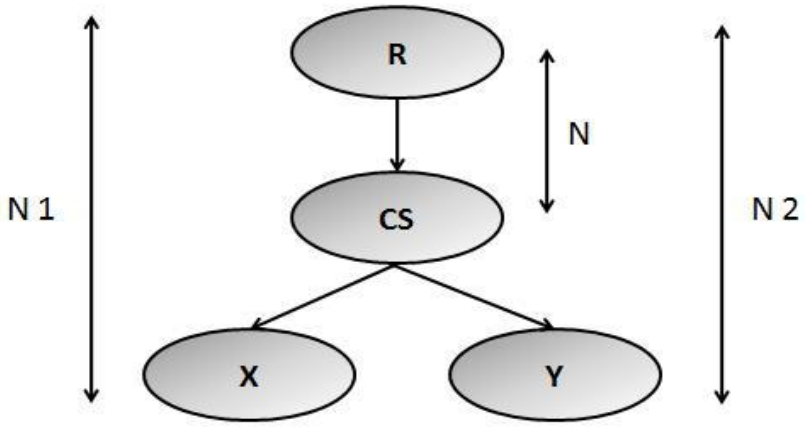

Figure 7. Exemple d'un extrait d'ontologie

La mesure de Wu et Palmer est définie par la formule suivante :

$$\boldsymbol{Sim\ (X,Y) = \frac{2 * N}{N1 + N2}}$$

5.1.2 Mesure de Rada et al (1989)

Cette mesure est adoptée dans un réseau sémantique et elle est fondée sur le fait qu'on peut calculer la similarité en se basant sur les liens hiérarchiques «is-a». Pour calculer la similarité de concepts dans une ontologie, on doit calculer le nombre des arcs minimums qui les séparent. Cette mesure [35], est basée sur le calcul de la distance entre les nœuds par le chemin le plus court, présente un moyen des plus évidents pour évaluer la similarité sémantique dans une ontologie hiérarchique. Il présente ainsi une mesure utilisant une métrique, distance (c1, c2), qui indique le nombre d'arcs minimum à parcourir pour aller d'un concept c1 à un concept c2:

$$\boldsymbol{Sim(c1, c2) = \frac{1}{1 + distance(c1, c2)}}$$

5.2 Approches basées sur les nœuds

Ces approches adoptent une nouvelle mesure en termes de la mesure entropique (Contenu Informationnel) de la théorie de l'information.

La probabilité P pour l'identification de l'utilisation d'une classe ou de ses descendants dans un corpus désigne l'information de la classe.

On définit l'entropie d'une classe par la formule suivante : $\boldsymbol{E(c) = - log\ (P(c))}$

Où P est la probabilité de trouver une instance du concept c.

La probabilité d'un concept c est calculée en divisant le nombre des instances de c par le nombre total des instances.

En associant des probabilités aux concepts d'une taxonomie, il est possible d'éviter le manque de fiabilité des distances des arcs. Cette caractéristique quantitative de l'information fournit

une nouvelle façon de mesurer la similarité sémantique. Plus l'information est partagée par deux concepts, plus ils sont similaires.
Parmi les travaux, recensés dans la littérature, sous cette catégorie on peut citer :

5.2.1 Resnik (1999)

Resnik [34] définit la similarité sémantique entre deux concepts par la quantité d'information qu'ils partagent. Cette information partagée est égale au contenu informationnel (CI) du plus petit généralisant (PPG) (Plus Petit Généralisant est le concept le plus spécifique qui subsume les deux concepts dans l'ontologie). Par exemple, dans la figure 7 le PPG des concepts X et Y est le concept CS

$$Sim\ (c1, c2) = Max\ [CI\ (PPG\ (c1, c2))] \quad \text{Où} \quad CI = -log\ (P\ (c))$$

5.2.2 Mesure de Lin (1998)

Lin a défini la similarité de concepts comme le rapport entre la quantité d'informations nécessaires pour indiquer le point commun entre ces deux concepts et les informations nécessaires pour les décrire [25].
Cette mesure est légèrement différente de celle de Resnik :

$$\boldsymbol{Sim}\ (\boldsymbol{a}, \boldsymbol{b}) = \frac{\mathbf{2} * \boldsymbol{CI}\ (\boldsymbol{PPG}(\boldsymbol{a}, \boldsymbol{b}))}{\boldsymbol{CI}\ (\boldsymbol{a}) + \boldsymbol{CI}\ (\boldsymbol{b})}$$

5.3 Approches hybrides

Ces approches sont fondées sur un modèle qui combine entre les approches basées sur les arcs (Distances) en plus du contenu informationnel qui est considéré comme facteur de décision.

5.3.1 Mesure de Jiang et Conrath (1997)

Pour remédier au problème présenté au niveau de la mesure de Resnik, Jiang et Conrath [22] ont apporté une nouvelle formule qui consiste à combiner l'entropie (contenu informationnel) du concept spécifique à ceux des concepts dont on cherche la similarité (combine entre les techniques basées sur les arcs et les techniques basées sur les nœuds qui consistent à compter les arcs afin d'améliorer les résultats par des calculs basés sur les nœuds). Notons que cette formule est définie par l'inverse de la distance sémantique.

$$\boldsymbol{Sim}\ (\boldsymbol{c1}, \boldsymbol{c2}) = \frac{\mathbf{1}}{\boldsymbol{distance}(\boldsymbol{c1}, \boldsymbol{c2})}$$

Sachant que la distance entre c1 et c2 est calculée par la formule suivante :

$$Distance\ (c1, c2) = CI\ (c1) + CI\ (c2) - (2.CI\ (PPG\ (c1, c2)))$$

5.3.2 Mesure de Leacock et Chodorow (1998)

Une autre méthode présentée combine la méthode de comptage des arcs et la méthode du contenu informationnel [35]. La mesure proposée par Leacock et Chodorow est basée sur la longueur du plus court chemin entre deux synsets de Wordnet. Les auteurs ont limité leur

attention à des liens hiérarchiques «is-a» ainsi que la longueur de chemin par la profondeur globale de la taxonomie. La formule est définie par :

$$\boldsymbol{Sim\ (X,Y) = -Log\left(\frac{CD(X,Y)}{2 * M}\right)}$$

Où M est la longueur du chemin le plus long qui sépare le concept racine, de l'ontologie, du concept le plus en bas (la profondeur globale). On dénote par CD (X, Y) la longueur du chemin le plus court qui sépare X de Y.

5.4 Approches basées sur l'espace vectoriel

Dans le domaine de la recherche de l'information, les modèles de l'espace vectoriel sont largement adoptés, on parlera alors de similarité numérique. Ces approches [35] utilisent un vecteur caractéristique, dans un espace dimensionnel, pour représenter chaque objet et calculent la similarité numérique en se basant sur la mesure de cosine ou la corrélation de Pearson. Le modèle de l'espace vectoriel est employé pour un arrangement des objets complexes en les représentants comme des vecteurs de k-dimensions.
La définition de la similarité numérique entre deux vecteurs d'objets est obtenue par leurs contenus internes. Parmi les approches citées dans la littérature on peut citer :

5.4.1 Similarité de Cosine

Cette mesure utilise la représentation vectorielle complète, c'est-à-dire la fréquence des objets (mots).
Deux objets (documents) sont similaires si leurs vecteurs sont confondus. Si deux objets ne sont pas similaires, leurs vecteurs forment un angle (X, Y) dont le cosinus représente la valeur de la similarité. La formule est définie par le rapport du produit scalaire des vecteurs x et y et le produit de la norme de x et de y.

$$\boldsymbol{Sim\ (X,Y) = \cos\ (X,Y) = \frac{X * Y}{\|X\|_2 * \|Y\|_2}}$$

La mesure de Cosine [35] quantifie donc la similarité numérique entre les deux vecteurs comme le cosinus de l'angle entre les deux vecteurs.

5.4.2 Similarité de Pearson

La mesure de similarité de Pearson est basée sur le calcul de la corrélation. Pour connaître le coefficient de corrélation liant deux séries $X(x_1, x_2,\ldots x_n)$ et $Y(y_1, y_2,\ldots y_n)$, on applique la formule suivante :

$$\text{Sim (x,y)} = r_p = \frac{\sum_{i=1}^{N}(x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

Si r vaut 0, les deux courbes ne sont pas corrélées et donc ne sont pas similaire. Les deux courbes sont d'autant mieux corrélées que r est loin de 0 (proche de -1 ou 1).

Avec:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$ est la moyenne de X et $$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$ est la moyenne de Y

6. Limites des systèmes de recommandation

Malgré leur popularité croissante, les systèmes de recommandation ont subi quelques ratés. Nous identifions quelques problèmes dont ils souffrent [24] :

- **Démarrage à froid.** Souvent, on se retrouve confronté au problème qu'un utilisateur ne soit comparable avec aucun autre. Ce problème est dû au fait que peu ou pas d'utilisateurs ont évalué un article donné, ou qu'un utilisateur donné a évalué très peu ou pas d'articles. Généralement, ce problème survient quand un nouvel utilisateur ou une nouvelle ressource est ajouté à la base de recommandation.

- **Masse critique.** Afin de former de meilleures communautés, le système exige un nombre suffisant d'évaluations en commun entre les utilisateurs pour les comparer entre eux. Par exemple, on ne peut pas conclure que deux personnes sont dans une même communauté si elles n'ont qu'une seule évaluation en commun. La constatation aujourd'hui est que malgré la taille énorme de l'ensemble des documents, achats, etc., dans les systèmes, le nombre des évaluations en commun entre utilisateurs risque d'être faible.

- **Principe d'induction.** Les systèmes de recommandation se basent sur le principe qu'un utilisateur qui a exhibé un comportement dans le passé tendra à exhiber un comportement semblable dans le futur. Cependant, ce principe n'est pas nécessairement vrai dans le contexte réel. En effet, un utilisateur peut changer complètement de domaine d'intérêt ou en avoir plusieurs. Pour faire face à ce problème, des techniques de dérive d'intérêt (Interest drift) ou de changement de contexte (Context shifts) ont vu le jour.

- **Centralisé ou distribué.** Les systèmes de recommandation, tels qu'Amazon, sont typiquement basés sur une architecture centralisée. Le moteur de recommandation centralisé permet de sauvegarder le profil de l'utilisateur et le calcul des recommandations dans un serveur central. Cependant, en dépit de leur popularité, les systèmes de recommandation centralisés souffrent de plusieurs problèmes : le coût, la robustesse, la sécurité, la portabilité, etc. Une des solutions à ces problèmes est de répartir le système. On croit qu'un système de recommandation distribué pourrait être conçu pour tirer profit de la puissance de calcul disponible sur les ordinateurs des utilisateurs.

- **Sécurité et crédibilité.** Les systèmes de recommandation ne peuvent pas empêcher les actes de tromperie. Il devient ainsi facile de se forger une nouvelle identité et de se livrer au vandalisme. Il est plus difficile de contrôler l'identité des utilisateurs et de pénaliser le comportement malveillant. Par conséquent, il est indispensable d'avoir des moyens permettant à chaque utilisateur de décider en quels utilisateurs et en quels contenus avoir confiance.

- **Protection de la vie privée.** Un autre problème qui touche les systèmes de recommandation est la protection des informations sensibles constituant le profil de l'utilisateur (informations personnelles, intérêts, goûts, habitudes, etc.). Vu la nature de l'information, ces systèmes doivent assurer une telle protection. Ainsi, des moyens, pour préserver l'anonymat des utilisateurs et chiffrer les données transmises, sont nécessaires.

7. Les principaux travaux dans les systèmes de recommandation

Dans le domaine de la recommandation, plusieurs techniques ont été utilisées par les chercheurs, notamment, les techniques basées sur les heuristiques et celles basées sur un modèle. Nous pouvons citer parmi ces techniques : les réseaux baésiens, le clustering, les arbres de décision, etc (voir réf [13]). Le tableau 4 résume les différentes techniques de recommandation utilisées par la communauté des chercheurs dans les trois approches de recommandation (FBC, FC et hybride).

Tableau 4. Classification des recherches dans les systèmes de recommandation

Recommendation Approach	Recommendation Technique	
	Heuristic-based	Model-based
Content-based	Commonly used techniques: • TF-IDF (information retrieval) • Clustering Representative research examples: • Lang 1995 • Balabanovic & Shoham 1997 • Pazzani & Billsus 1997	Commonly used techniques: • Bayesian classifiers • Clustering • Decision trees • Artificial neural networks Representative research examples: • Pazzani & Billsus 1997 • Mooney et al. 1998 • Mooney & Roy 1999 • Billsus & Pazzani 1999, 2000 • Zhang et al. 2002
Collaborative	Commonly used techniques: • Nearest neighbor (cosine, correlation) • Clustering • Graph theory Representative research examples: • Resnick et al. 1994 • Hill et al. 1995 • Shardanand & Maes 1995 • Breese et al. 1998 • Nakamura & Abe 1998 • Aggarwal et al. 1999 • Delgado & Ishii 1999 • Pennock & Horwitz 1999 • Sarwar et al. 2001	Commonly used techniques: • Bayesian networks • Clustering • Artificial neural networks • Linear regression • Probablistic models Representative research examples: • Billsus & Pazzani 1998 • Breese et al. 1998 • Ungar & Foster 1998 • Chien & George 1999 • Getoor & Sahami 1999 • Pennock & Horwitz 1999 • Goldberg et al. 2001 • Kumar et al. 2001 • Pavlov & Pennock 2002 • Shani et al. 2002 • Yu et al. 2002, 2004 • Hofmann 2003, 2004 • Marlin 2003 • Si & Jin 2003
Hybrid	Combining content-based and collaborative components using: • Linear combination of predicted ratings • Various voting schemes • Incorporating one component as a part of the heuristic for the other Representative research examples: • Balabanovic & Shoham 1997 • Claypool et al. 1999 • Good et al. 1999 • Pazzani 1999 • Billsus & Pazzani 2000 • Tran & Cohen 2000 • Melville et al. 2002	Combining content-based and collaborative components by: • Incorporating one component as a part of the model for the other • Building one unifying model Representative research examples: • Basu et al. 1998 • Condliff et al. 1999 • Soboroff & Nicholas 1999 • Ansari et al. 2000 • Popescul et al. 2001 • Schein et al. 2002

8. Conclusion

Dans ce chapitre, nous avons d'abord, présenté la notion des systèmes de recommandation, en détaillant les trois approches les plus utilisées, à savoir, l'approche FBC, FC et hybride. Ensuite, nous avons défini la notion de profil utilisateur avec ses deux facettes explicite et implicite. Nous avons également passé en revue les différentes classes de mesures de similarité utilisées par les SRs pour faire le matching entre deux profils utilisateur, deux contenus, ou un profil utilisateur et un descripteur de contenu. Enfin, nous avons terminé en citant quelques problèmes rencontrés par les systèmes de recommandation classiques. Parmi les problèmes évoqués est le fait que ces systèmes ne tiennent pas compte du contexte dans lequel se trouve l'utilisateur lorsqu'il désire une recommandation. Les systèmes de recommandation sensibles au contexte (les CARS : Context Aware Recommender Systems) apportent une solution à ce problème. Nous les détaillons dans le chapitre suivant.

CHAPITRE II

Les systèmes de recommandation sensibles aux contextes

1. Introduction

Les systèmes de recommandation jouent un rôle important dans la manipulation de grandes quantités d'informations. Souvent, le contenu et les objets qui pourraient intéresser une personne, dépendent de sa situation spécifique: l'emplacement actuel, la saison, le rôle de l'utilisateur, la température, etc. Les systèmes de recommandation sensibles aux contextes tentent d'exploiter l'utilisation du contexte afin d'améliorer le processus de génération des recommandations.

Dans ce chapitre, nous commençons d'abord par donner une définition de la notion du contexte, son importance, ses dimensions, et la pertinence de son choix. Ensuite nous enchainons en présentant les différentes méthodes permettant l'introduction du contexte ainsi que les différents travaux sur les systèmes de recommandation sensibles aux contextes. Enfin, nous terminons en comparant ces différents systèmes.

2. Définition de contexte

Il y a plus de 150 définitions de diverses disciplines, l'une des plus fréquemment citées a été proposée par Abowd, Dey et al [1]:

"Le contexte est n'importe quelle information qui peut être utilisée pour caractériser une situation d'une entité. Une entité peut être une personne, un endroit, ou un objet que l'on considère comme étant pertinent à l'interaction entre un utilisateur et une application y compris ces deux derniers"

Exemple : Dans la figure 8, l'attribut contextuel K est "la période de l'année", dans laquelle un client peut faire des achats sur le portail e-commerce. La classification de l'attribut k est comme suit : l'été (Juin, Juillet, Août, Avril, Mai et Septembre), l'hiver (Octobre, Novembre, Décembre, Janvier, Février et Mars), en vacances (les trois premiers mois) ou pas en vacances (le reste des mois). En s'accordant à cette définition, un achat effectué le 1er Décembre est étiqueté en "vacances d'hiver". Cet attribut contextuel K peut être utilisé aussi dans une application de recommandation de voyages. Par exemple, les voyages recommandés durant les vacances d'été ne sont pas les mêmes que ceux recommandés pendant les vacances d'hiver.

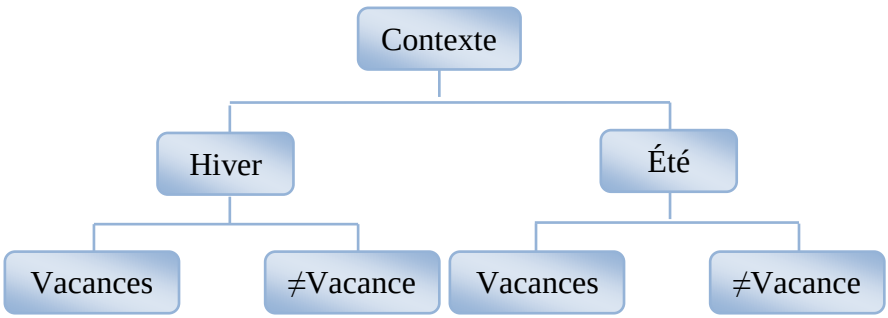

Figure 8. Exemple de contexte « période de l'année »

3. Importance du contexte dans les systèmes de recommandation

Dans la plupart des systèmes de recommandation personnalisés, certains problèmes peuvent survenir, comme par exemple, recommander des produits inappropriés à un utilisateur, qui effectue un achat d'un cadeau sur internet pour l'offrir à son ami. Le système risque de lui recommander des produits similaires, alors qu'il ne s'agit pas de ses propres préférences.

Ce problème peut être surmonté si le système connait " le contexte " dans lequel la transaction a eu lieu. Bien qu'il ne soit pas évident d'obtenir ce genre d'information, il est possible de la déduire des données dont on dispose d'une façon explicite (demande à l'utilisateur de spécifier les variables contextuelles) ou implicite (comme par exemple remarquer si le lieu de résidence d'un utilisateur a changé, ou bien, quand est-ce qu'il a effectué les transactions à travers ses fichiers journaux).

D'après S. Lombardi et al. [27], le modèle contextuel donne toujours de meilleurs résultats par rapport au modèle non-contextuel en termes de précision (nombre de cas correctement classés parmi tous les cas), et de rappel (nombre de tout les cas estimés pertinent par l'utilisateur et qui sont prévus correctement).

4. Dimensions du contexte

M. Daoud [15], a proposé dans le cadre de son travail deux dimensions principales du contexte de recherche (le contexte de l'utilisateur et le contexte de la requête). Elle vise à intégrer ces deux dimensions du contexte dans un processus de recherche d'information (ou recommandation) afin d'asseoir un système de recherche d'information contextuel. La première permet de caractériser l'utilisateur à travers ses centres d'intérêts à court et à long terme et ses préférences. La seconde dimension est liée à la requête et permet de cibler la nature des résultats attendus par l'utilisateur.

4.1. Contexte lié à l'utilisateur

On définit le contexte lié à l'utilisateur par des éléments qui peuvent être donnés explicitement par l'utilisateur ou déduits implicitement par le système. On cite les centres d'intérêts qui peuvent appartenir aux domaines de la science, sport, humanité, etc., et les préférences liées à la qualité des informations attendues. Ces préférences sont données explicitement et comportent les deux aspects suivants :

- La fraîcheur : l'utilisateur peut s'intéresser à l'information la plus récente ou bien à des documents liés à des dates bien précises,
- La granularité : l'utilisateur peut s'intéresser à un certain niveau de détail de la réponse attendue, cela peut déterminer la structure de la réponse retournée : un document intégral ou une partie de document (passage), une page d'accueil ou une page cible.

4.2. Contexte lié à la requête

On définit ce contexte par différents paramètres descriptifs et éventuellement mesurables à partir de la requête ou à partir du profil des n top documents retournés par la requête. Soient les paramètres suivants :

- Clarté de la requête : ce paramètre quantifie le degré d'ambiguïté de la requête et permet de mieux cibler la recherche,

- Distribution de la datation des documents : c'est la distribution des n meilleurs documents selon le paramètre date, et qui permet de mettre en évidence la fraîcheur de l'information,
- Degré de couverture de la requête par les services web : ce paramètre permet d'orienter la réponse retournée vers un document ou service.

5. Comment obtenir le contexte ?

Une question cruciale est de savoir comment obtenir les informations contextuelles riches [3]. Dans certaines circonstances, le contexte est explicite, comme une personne qui communique au système de recommandation les films qu'elle désire regarder.

D'autre part, l'information contextuelle peut également être déduite par le système. Par exemple, le temps qui peut être déduit à partir de l'horloge du système ou la localité de l'utilisateur qui peut être déduite à partir de son GPS. Il peut être déduit aussi en observant le comportement d'une personne qui a acheté un objet à partir d'un site Web e-commerce, pendant un jour de semaine ou un week-end. On peut également utiliser des informations spécifiques au domaine, qui sont généralement collectées à partir du CMS (Système de Gestion du Contenu). Par exemple, si un élément représente un accès à une musique, le genre de la musique peut être utilisé comme une dimension de l'information contextuelle.

6. Pertinence de l'information contextuelle

Les informations contextuelles ne sont pas toutes pertinentes pour la formulation des recommandations [3], par exemple, quelle est l'information contextuelle pertinente au moment de recommander un livre?

- Dans quel but le livre est acheté? (travail, loisirs, ...)
- Quand le livre sera lu? (semaine, week-end, ...)
- Où le livre sera lu? (A la maison, à l'école, dans un avion, ...)

La détermination de la pertinence des informations contextuelles se fait :

- manuellement, par exemple, à l'aide de la connaissance du domaine d'un concepteur du système de recommandation
- automatiquement, par exemple, en utilisant des procédures de sélection de l'apprentissage automatique [23], de l'exploration de données [26], et des statistiques [14].

7. Différentes méthodes d'introduction du contexte

Selon U. Panniello et al. [32], trois différents paradigmes algorithmiques pour l'intégration des informations contextuelles dans le processus de recommandation sont présentés:

7.1 L'approche de Pré-filtrage Contextuel (Pref)

Dans l'approche de pré-filtrage, l'information contextuelle est utilisée comme une étiquette permettant de filtrer les évaluations qui ne correspondent pas à l'information contextuelle spécifiée. Cela se fait avant que la méthode principale de recommandation soit lancée sur le reste des données sélectionnées. Autrement dit, si un contexte d'intérêt particulier est k, alors cette méthode sélectionne à partir de la série initiale toutes les évaluations relatives au

contexte spécifié k, et elle génère la matrice « articles x utilisateur » ne contenant que les données relatives au contexte k. Après cela, la méthode des systèmes de recommandation (SR), comme le filtrage collaboratif, est lancée sur la base de données réduite afin d'obtenir les recommandations liées au contexte k. La figure 9 montre l'intégration du contexte dans un système de recommandation à deux dimensions (utilisateur x produit).

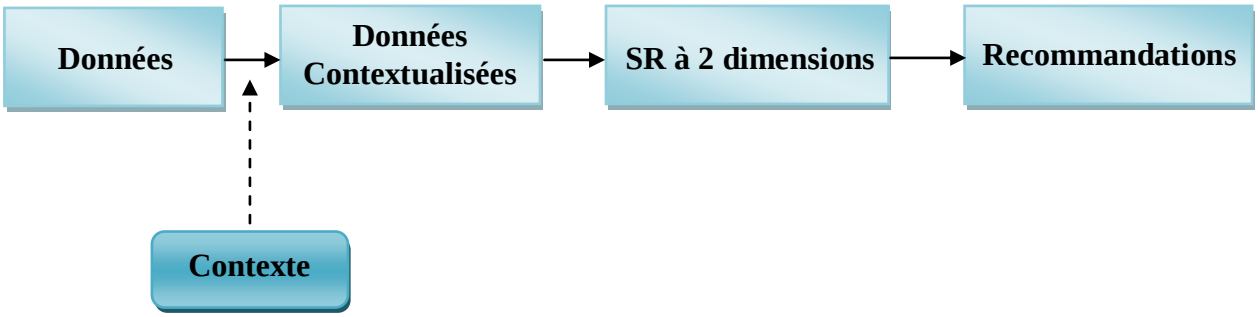

Figure 9. Pré-filtrage Contextuel

Il existe différents types de pré-filtrage, tels que le pré-filtrage exact (Exact Pre-Filtering) et le pré-filtrage généralisé (Generalized Pre-Filtering). Exact Pre-Filtering (EPF) sélectionne toutes les opérations relatives exactement au contexte spécifique (par exemple, les achats effectués en hiver 2008), tandis que Generalized Pre-Filtering sélectionne toutes les opérations visées à un contexte spécifique basée sur la généralisation de l'information contextuelle (par exemple, les achats effectués en "hiver 2008, pas en vacances" peut être généralisée aux achats effectués en "2008 pas en vacances", ou un "Cadeau pour ses parents" peut être généralisée à "cadeau pour ses relations"). Il est à noter qu'il existe différentes possibilités pour cette généralisation basée sur le contexte taxonomique / granularité, alors que Exact Pre-Filtering est défini uniquement par le contexte k.

7.2 L'approche de Post-filtrage Contextuel (PoF)

Selon l'approche post-filtrage, l'information contextuelle est utilisée après le lancement de la méthode principale de recommandation 2D (à deux dimensions). Une fois les évaluations inconnues sont estimées et les recommandations sont produites, le système analyse les données pour un utilisateur donné dans un contexte précis pour trouver les modèles d'utilisation des articles spécifiques, et utilise ces modèles pour «contextualiser» les recommandations obtenues à partir de la méthode classique de recommandation 2D, comme le filtrage collaboratif. La figure 10 montre comment utiliser le contexte après la recommandation (2D).

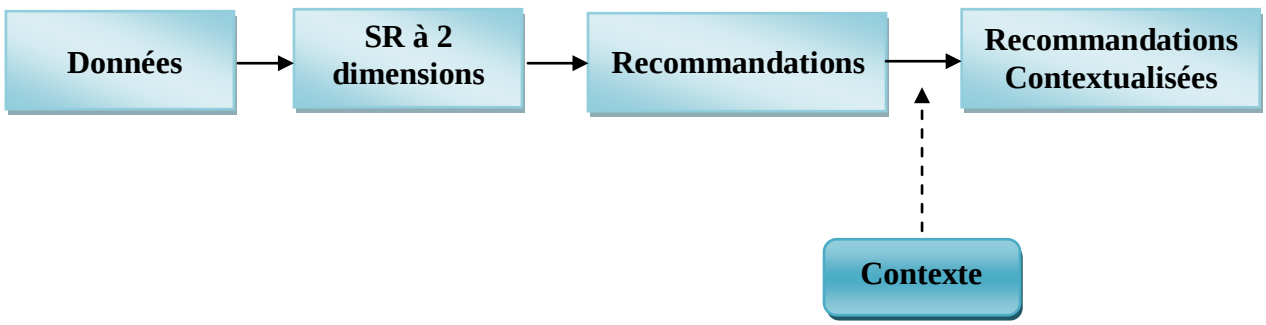

Figure 10. Post-filtrage Contextuel

Il existe différentes méthodes pour la contextualisation des recommandations 2D. Dans ce qui suit, nous nous concentrons sur les deux approches appelées "Weight" et "Filter". Les deux approches analysent les données pour un utilisateur dans un contexte donné pour calculer la probabilité contextuelle avec laquelle l'utilisateur choisit un certain article dans un contexte donné. Les recommandations obtenues en utilisant la méthode 2D sont «contextualisées» en utilisant les probabilités contextuelles. La probabilité contextuelle P_k (i, j), avec laquelle le i-ème client achète le j-ème élément dans le contexte k, est calculée comme le nombre de voisins (les clients similaires à i) qui ont acheté le même article dans le même contexte divisée par le nombre total de voisins.
Les approches diffèrent selon la façon avec laquelle les recommandations sont contextualisées. Le Weight PoF multiplie chaque évaluation 2D par $P_k\ (i, j)$:

$$\boldsymbol{Rating_k\ (i,j) = Rating \times P_k\ (i,j)}$$

Tandis que Filter PoF filtre les évaluations 2D à partir d'une valeur seuil de $P_k\ (i, j)$:

$$\boldsymbol{Rating_k\ (i,j) = \begin{cases} Rating\ (i,j) & if \quad P_k\ (i,j) \geq P^* \\ 0 & if \quad P_k\ (i,j) < P^* \end{cases}}$$

Où P^* est la valeur du seuil.

U. Panniello, A. Tuzhilin, M. Gorgoglione [32], ont comparé les approches pré- filtrage et post-filtrage pour générer des recommandations contextuelles. En particulier, ils ont utilisé la méthode Exact Pre-Filtering (EPF) et les méthodes de Post-filtrage Weight et Filter. Ils ont comparé ces méthodes à travers des ensembles de données, et les expérimentations ont montré que la méthode du Filter domine EPF et que l'EPF domine la méthode du Weight. Cela implique que la comparaison entre les approches de pré et post-filtrage dépend de manière significative du type de la méthode de post-filtrage utilisée.

7.3 L'approche de la Modélisation Contextuelle (CM)

En plus des deux méthodes citées auparavant, il existe une troisième méthode à savoir la modélisation contextuelle dans laquelle l'information contextuelle est utilisée directement à l'intérieur des algorithmes de génération des recommandations. La figure 11 montre l'intégration du contexte dans le processus de recommandation.

Figure 11. Modélisation contextuelle

8. Les différents travaux sur les systèmes de recommandation sensibles au contexte

Plusieurs travaux ont été effectués sur les systèmes de recommandation introduisant la notion du contexte. Nous décrivons dans une première sous-section un certain nombre de ces systèmes avant de comparer leurs propriétés dans une seconde sous-section. Nous avons présenté la synthèse ainsi que la comparaison de ces travaux dans [38].

8.1 Exemples de SR sensibles au contexte

Nous présentons, dans cette section, quelques systèmes de recommandation sensibles au contexte.

8.1.1 Micro-profiling

L'approche micro-profiling [8] a pour but de recommander des chansons/artistes inconnus à un utilisateur. Le type du système de recommandation est le pre-filtering car les données sont filtrées selon le contexte (le temps : moment de la journée, jour, mois ou année) dans cette approche avant d'effectuer les recommandations.

Dans cette approche, on suppose que les goûts des utilisateurs changent selon une période et peuvent être similaires dans une même période. Par exemple, un utilisateur préfère écouter un certain genre de chanson quand il travaille, et un autre genre de chanson avant de dormir.

On dispose d'un profil utilisateur dans lequel on affectera les chansons/artistes écoutés par cet utilisateur. Ce profil sera décomposé de façon à ce qu'il s'adapte à chaque contexte : dans cette approche, le temps représente le moment de la journée, il sera divisé en plusieurs segments en fonction des chansons que l'utilisateur écoute le plus durant chaque moment (segment de temps) de la journée.

Le problème qui se pose par cette approche est, comment répartir les heures de la journée ? C'est-à-dire comment bien diviser le temps en segments, car la définition de la matinée peut changer d'un utilisateur à un autre. Donc l'erreur du partitionnement du temps doit être minimale.

Les contextes sont extraits implicitement, ce qui rend la tâche assez difficile car on ne connaît pas ce que l'utilisateur n'aime pas. On ne peut déduire que ce qu'il aime à partir de ce qu'il a écouté comme albums d'artistes ou chansons et à quel moment de la journée il les a écoutés.

Pour estimer l'erreur commise (se fait en hors-ligne), les prédictions des préférences générées sont comparées aux préférences des utilisateurs obtenues explicitement.

Les différentes étapes de la méthode :

- Extraction des données implicites (les préférences) et les diviser en plusieurs segments contextualisés.
- Transformer chaque segment en matrice A(*utilisateur x item)* où A(*u, i*) correspond aux préférences explicites.

L'utilisation de ces micro-profiles améliore l'exactitude de la recommandation. L'un des problèmes posés par cette méthode, réside dans le fait que le partitionnement du temps diffère

d'un utilisateur à un autre (au lieu d'être fixe pour tous les utilisateurs), ce qui risque de diminuer la précision du système.

8.1.2 DaVI

L'approche DaVI (Dimension as virtual items) [17] introduit la notion du contexte qui sera considérée comme étant un item virtuel structuré de la même manière que les autres items. Par conséquent, il sera traité par les différentes approches de recommandations de la même façon que les autres articles. L'approche utilise l'un des deux algorithmes, à savoir, le filtrage basé sur le contenu et les règles d'associations.

Le contexte sera obtenu à partir des données multidimensionnelles : fichiers journaux des sites web (identificateur de l'utilisateur, page visitée, date, heure...) qui sont sauvegardés dans la base de données. L'exemple ci-dessous présente les lignes d'un fichier utilisant les champs suivants : Heure, Adresse IP du client, Méthode, Ressource URI, État HTTP et Version HTTP.

#Software: Microsoft Internet Information Services 5.0
#Version: 1.0
Date 2011-01-01 17:42:15
#Fields: time c-ip cs-method cs-uri-stem sc-status cs-version
17:42:15 172.16.255.255 GET /default.htm 200 HTTP/1.0

Cette entrée indique que le 1 janvier 2011 à 17h42, un utilisateur disposant de HTTP version 1.0 et dont l'adresse IP est 172.16.255.255 a lancé une commande HTTP GET pour le fichier Default.htm. La demande est revenue sans erreur. Le champ #Date: indique la date à laquelle la première entrée de journal a été effectuée. Cette date correspond à la création du fichier journal. Le champ #Version: indique que le format de fichier journal W3C est utilisé.

Tous les champs peuvent être sélectionnés, mais certains champs peuvent ne contenir aucune information pour certaines demandes. Les champs sélectionnés ne comportant aucune information affichent un tiret (—) comme marque de réservation.

Ce contexte sera regroupé en deux parties :

- le contexte obtenu par les fichiers journaux.
- le contexte obtenu par le CMS (content management systems).

On dispose d'un ensemble I qui contient tous les items du site web. A partir du filtrage collaboratif basé sur les articles (Item-based CF), on obtient la relation (grâce au calcul de similarité) entre les items dans I.

Tableau 5. Matrice de similarité article-article

	$\mathbf{i_1}$	$\mathbf{i_2}$	…….	$\mathbf{i_k}$
$\mathbf{i_1}$	1	$Sim(i_1, i_2)$	…….	$Sim(i_1, i_k)$
$\mathbf{i_2}$	$Sim(i_2, i_1)$	1	…….	$Sim(i_2, i_k)$
…….	……..	…….	1	…….
$\mathbf{i_k}$	$Sim(i_k, i_1)$	$Sim(i_k, i_2)$	…….	1

Une partie O de l'ensemble I représente l'historique des accès faits par l'utilisateur (représentant par exemple l'ensemble des chansons écoutées par un individu); le système recommande les items les plus similaires à ceux de l'ensemble O.

Appliquer l'approche DaVI sur l'algorithme Item-Based CF consiste à ajouter des lignes et des colonnes pour chaque élément de contexte dans la matrice de similarité. De ce fait, l'introduction du contexte servira à recommander un item ressemblant à celui qui a été consulté dans un contexte donné. Par exemple, si on s'intéresse au contexte date et plus exactement au contexte jours de la semaine, on pourrait recommander pour un individu qui a regardé un film d'horreur par un Lundi le même genre de film tous les Lundi des semaines à venir (en déduisant que cet individu aime regarder les films d'horreur tous les Lundi). Cela permettra d'améliorer la qualité des items recommandés.

Tableau 6. Matrice de similarité avec l'attribut contextuel jour

	i_1	i_2	…….	i_k	j_1	…..	j_n
i_1	1	$Sim(i_1, i_2)$	…….	$Sim(i_1, i_k)$	$Sim(i_1, j_1)$	…..	$Sim(i_1, j_n)$
i_2	$Sim(i_2, i_1)$	1	…….	$Sim(i_2, i_k)$	$Sim(i_2, j_1)$	…..	$Sim(i_2, j_n)$
…….	…….	…….	1	…….	……	…..	…..
i_k	$Sim(i_k, i_1)$	$Sim(i_k, i_2)$	…….	1	$Sim(i_k, j_1)$	…..	$Sim(i_k, j_n)$
j_1	$Sim(j_1, i_1)$	$Sim(i_1, i_2)$	……	$Sim(j_1, i_k)$	1	…..	$Sim(j_1, j_n)$
……..	……..	……	……..	…….	….	1	….
j_n	$Sim(j_n, i_1)$	$Sim(j_n, i_2)$	……	$Sim(j_n, i_k)$	$Sim(j_n, j_1)$	…..	1

L'application de DaVI aux règles d'associations (AR) : il s'agit ici, de déduire un ensemble d'items à recommander (pour un individu donné) à partir d'un ensemble de sessions web présentées sous forme de paires <identifiant de l'utilisateur, item> en utilisant des règles d'association.

8.1.3 DISCOVR

Le système DISCOVR [20] (voir figure 12) a été développé sous forme de services qui sont configurables via des fichiers XML. Ces services peuvent être combinés de façon à ce que la sortie de l'un constitue l'entrée de l'autre, et ce durant tout le processus de recommandation. Ceci facilitera l'intégration de plusieurs techniques de recommandation, que ce soit en parallèle ou séparément et avec ou sans la prise en considération du contexte.

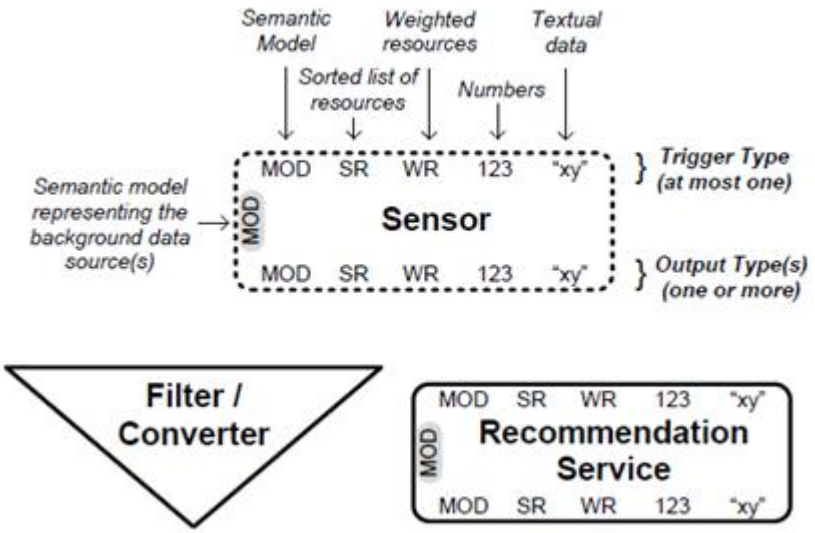

Figure 12. Types des composants de l'architecture DISCOVER.

MOD : Modèle sémantique.
SR : Listes des ressources.
WR : Le poids des ressources inclut entre 0 (pour les plus faibles) et 1 (pour les plus importants).
123 : Chiffres.
''XY'' : Données de type texte.

On peut combiner ces services entre eux : utiliser la sortie d'un service comme entrée d'un autre service durant tout le processus de recommandation. Ceci posera un problème : le service destinataire doit connaître beaucoup de chose sur l'implémentation interne du service source pour pouvoir bien interpréter ses sorties.

Il existe 3 types de service dans le système DISCOVR:

- *Sensor :* Retourne le contexte sous forme de poids de ressources ou de donnée numérique (par exemple en introduisant l'adresse IP de l'utilisateur dans le site « www.mon-ip.com/localiser-adresse-ip.php », on peut déduire la ville où il habite)
- *Service de recommandation :* Produit les recommandations et retourne une liste ordonnée d'items qui sont susceptibles de plaire à l'utilisateur et retourne aussi une représentation des données utilisateurs sous forme de modèle sémantique.
- *Utilités :* sont des modules supplémentaires pour filtrer et/ou convertir des données : filtrer pour ne recommander que les données pertinentes et convertir les entrées de chaque service pour qu'elles lui correspondent.

L'ordre d'exécution des services est géré par un graphe orienté qui est créé par le "service processor". La sortie de chaque service est traitée par la composante "view preparer".
Le résultat est intégré dans la plateforme JSP afin de l'afficher d'une manière lisible à l'utilisateur. Voir figure 13.

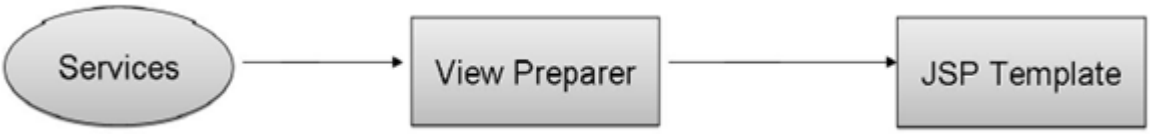

Figure 13. Etape de traitement des services

L'avantage de cette méthode (Discover) est d'offrir une architecture extensible facile à intégrer dans les systèmes de recommandation et de prototyper d'une manière rapide grâce au traitement qui peut se faire en parallèle.

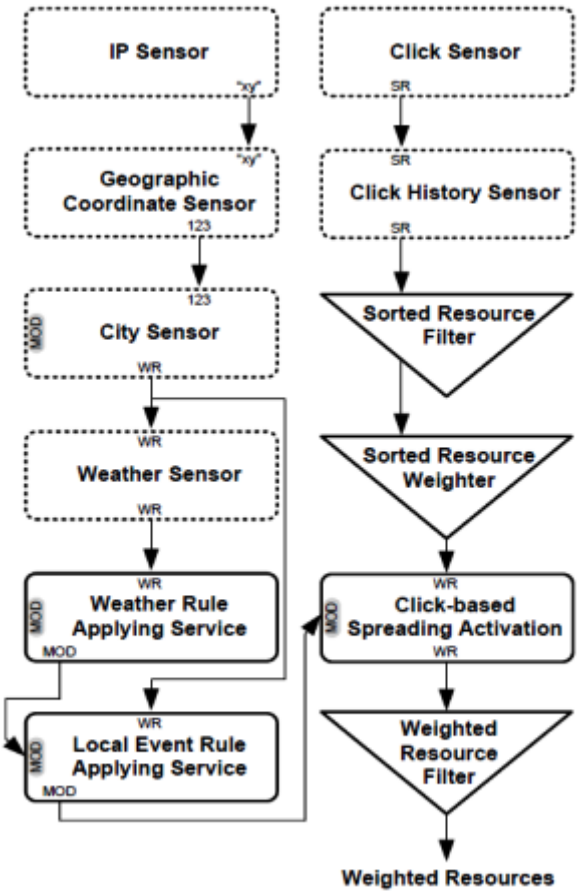

Figure 14. Exemple d'interaction de plusieurs services

La figure 14 montre la création des recommandations pour des événements, qui sont :
a) prenant place dans (ou près de) la ville de l'utilisateur,
b) qui sont adaptés aux conditions météorologiques actuelles et,
c) reflète les préférences générales de l'utilisateur, comme son goût de musique ou son club de sport préféré, que l'on déduit des cliques de l'utilisateur.

Comme on le voit dans le schéma, le processus de recommandation examiné à en effet deux points de départ qui démontre que ces processus ne doit pas être linéaire. Dans la branche de gauche, le capteur IP récupère l'adresse IP de l'utilisateur courant et la retourne comme une chaîne de caractères. Le capteur des coordonnées géographiques utilise cette adresse IP pour déterminer l'emplacement actuel de l'utilisateur (via un service Web accessible au public).

Les coordonnées géographiques résultant déclenchent le détecteur de la ville, qui cherche maintenant les villes les plus proches en comparant les coordonnées de ceux qui sont stockés dans le modèle du domaine. Le résultat est une liste de ressources pondérés ('Oran→ 1.0' ,'Tlemcen → 0.2', ce qui signifie que les événements d'Oran sont hautement recommandable par rapport aux événements de Tlemcen).

Maintenant qu'on a les villes les plus proches à l'utilisateur, ce résultat est introduit dans deux autres services: Le capteur météorologique utilise la ville la plus proche et détermine le climat à cet endroit (en utilisant un autre service web).

Par la suite, la branche de droite (figure 14) est en cours: Les produits et les événements sur lesquels l'utilisateur a cliqué sont obtenus en utilisant le capteur Cliquer et en utilisant le service « historique de clicks ». A chaque fois qu'un utilisateur clique sur une page, le système sauvegarde cette page et lui affecte un poids (temps écoulé depuis le click jusqu'à qu'il quitte la page en question). Après cela, le système filtre ces clicks en limitant le nombre d'éléments et les poids qui leur sont attribués. Ensuite il déclenche un processus de recommandation basé sur l'activation de la diffusion (c'est-à-dire il recommande des pages similaires à celles que l'utilisateur préfère). Ce processus prend le modèle sémantique de la branche de gauche comme une source de données (c'est-à-dire il fonctionne sur le modèle contenant les événements locaux du climat actuel).

Enfin, la liste des ressources pondérée peut être limitée au dix plus importants articles par exemple. La liste finale des ressources pondérées est maintenant en bonne forme et affichée à l'utilisateur.

8.1.4New@hand

Le système News @ Hand [12] est un système de recommandation sensible au contexte. La figure 15 montre l'interface graphique de ce système. Dans New @ hand une notion particulière de contexte à été proposée à savoir: le contexte d'exécution sémantique, C_u^t sous lequel les activités d'un utilisateur u se produisent au sein d'une unité de temps donnée t. Une fois le contexte est construit, une activation contextuelle des préférences est déclenchée en trouvant les parcours sémantiques reliant les préférences au contexte.

Ce système de recommandation utilise les ontologies (technologie sémantique) qui peut être représentée sous forme de plusieurs domaines : Sport, santé, science, business…etc.
Si l'utilisateur s'est identifié, les recommandations personnalisées sont activées, sinon, des items lui sont recommandés au hasard.

Figure 15. L'interface graphique du système News @ Hand

Les préférences de l'utilisateur sont représentées sous forme de vecteur $u_m=(u_{m,1},\ldots, u_{m,k})$, où $u_{m,k} \in [-1,1]$ est le poids qui mesure l'intensité de l'intérêt que porte l'utilisateur u_m pour le concept C_k. Ce dernier appartient à une classe dans un domaine de l'ontologie et le k = le nombre total de concepts dans cette ontologie.

Si le poids est proche de 1, alors l'utilisateur aime le concept et s'il est proche de -1, l'utilisateur ne l'aime pas.

Les items sont représentés sous forme de vecteurs $i_n = (i_{n,1},\ldots, i_{n,k})$ avec $i_{n,k}$ le poids du concept $\in [0,1]$ qui mesure l'importance d'un concept dans le contenu d'un item.

L'intérêt pour les items peut être calculé en comparant les préférences de l'utilisateur avec les vecteurs des items.

Les avantages d'utiliser des ontologies au lieu d'utiliser l'extraction/recherche par mots clés sont :
• L'ontologie est plus précise et moins ambigüe.
• L'ontologie est hiérarchisée : Facilite les recherches et donc gagner plus de temps.
• L'inférence aide à trouver les relations entre les préférences de l'utilisateur.

L'introduction du contexte conduit très souvent au problème de « data sparsity ». Afin d'éviter ce problème dans le profil utilisateur, il faut trouver un rapprochement ou un lien entre les classes (concepts) des ontologies qui sont pertinentes pour l'utilisateur. Mais aussi, trouver des liens entre les concepts qui ne sont pas pertinents pour l'utilisateur.

L'extraction personnalisée du contenu (trouver les items pertinents) se fait en appliquant la fonction de cosin suivante :

$$\boldsymbol{Pref\ (i_n,u_m) = i_n \times u_m \,/\, ll\ i_n ll \times ll u_m ll}$$

Le résultat de la formule Є [0,1]. Si le résultat est proche de 0 donc les deux vecteurs sont différents. S'ils sont proches de 1, donc les deux vecteurs sont semblables et il est possible de les représenter par un seul vecteur.

La personnalisation doit combiner les préférences à long terme basées sur l'historique des préférences de l'utilisateur avec des prédictions contextualisées basées sur les activités courantes de l'utilisateur (feedback de l'utilisateur).

Cette personnalisation aide les utilisateurs à trouver les articles qui leurs sont pertinents. L'expansion sémantique des préférences de l'utilisateur assouplit l'appariement entre le profil utilisateur et l'item, en évitant les problèmes de data sparsity et le démarrage à froid.

La figure 16 montre l'expansion des préférences utilisateur, l'expansion du contexte et l'intersection des ces deux expansion qui donne les préférences contextualisées de l'utilisateur.

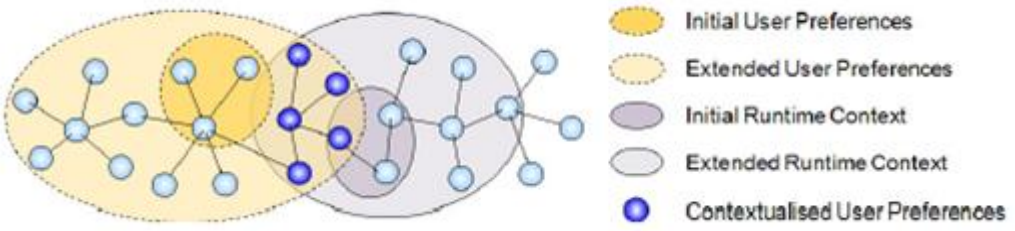

Figure 16. Expansion et contextualisation des préférences de l'utilisateur

8.1.5 Extention de My Map

MyMap [16] est un système de recommandation mobile (c'est-à-dire un système de recommandation qui peut être installé sur un appareil mobile comme un PDA, IPhone, ect.) qui est basé sur l'idée d'utiliser une carte afin de fournir des recommandations personnalisées de lieux d'intérêts pour les touristes.

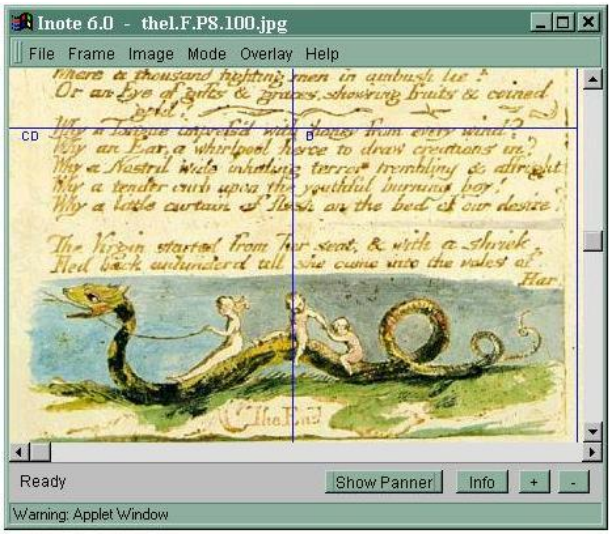

Figure 17. Imprime écran du logiciel Inote

La carte dans MyMap a été annotée avec l'outil graphique Inote [21] (voir figure 17) qui fournit un moyen d'annoter des images d'une manière conviviale, pour définir une correspondance entre ses objets graphiques et les métadonnées qui vont être utilisés par le système afin de générer des recommandations. La figure 18 montre un fragment de métadonnée décrivant un restaurant nommé Città Vecchia.

```
<overlay><title>bari-zone1</title>
<detail><title>eating</title>
      <annotation><title>eating_type</title>
      <text>restaurant</text> </annotation>
      <annotation><title>name</title>
      <text>Città Vecchia</text> </annotation>
      <annotation><title>coordinates</title>
      <text>41°06'14.800"N 16°45'57.013"E</text>
      </annotation>
      <annotation><title>food-type</title>
      <text>typical</text> </annotation>
      <annotation><title>view</title>
      <text>historical center</text></annotation>
      <annotation><title>table</title>
      <text>open-air</text></annotation>
      <annotation><title>timetable</title>
      <text>12.00-22.00</text></annotation>
      <annotation><title>closed</title>
      <text>Monday</text></annotation>
      <annotation><title>wheelchair accessibility</title>
      <text>yes </text></annotation>
            …
</detail>…
</overlay>
```

Figure 18. Fragment de métadonnée décrivant un restaurant

MyMap décide laquelle des informations qu'il faut fournir et comment la présenter, à partir d'une représentation XML des connaissances du domaine. Pour cet objectif, le système utilise deux composants:

- *le gestionnaire Mobile User Profile (MUP)* : utilise des profils formalisés (voir figure 19) selon le langage UbisWorld qui est une approche intéressante pour modéliser une partie du monde réel comme une ville et s'adapter dynamiquement a l'accès aux sources d'information (voir http://www.w2m.org).
- *le module de sélection et de présentation de l'information* : qui a comme tâche la génération des descriptions d'articles choisis, suite à une demande explicite de l'utilisateur ou de façon proactive en présence d'objets intéressants.

```
<Statement id="16">
<content> <subject><UbisWorld:Dora /></subject>
<predicate><UserOL:eating</predicate>
<predicate-range><UserOL: restaurant,fast-food,pizzeria/>
</predicate-range><object>restaurant</object>
<predicate><UserOL:table/></predicate>
<predicate-range><UserOL:open-air,inside..../>
</predicate-range><object>open-air</object>
</content>
<restriction><season>summer</season></restriction>
<meta>
<owner><UbisWorld:Dora /></owner>
<privacy><UbisWorld:friends /></privacy>
<purpose><UbisWorld:commercial /></purpose>
<retention><UbisWorld:short /></retention>
<viewer><UbisWorld:MyMap /></viewer>
<explanation confidence="0,75"  creator="Dora" evidence="
Interface input "method="acquire_pref" />
</meta>
</Statement>
```

Figure 19. Un fragment MUP d'une personne nommé Dora

Imaginons le scénario suivant : Dora (une touriste) est à Bari (ville italienne) pour la première fois. Quand elle est à l'étranger, elle préfère goûter des aliments locaux dans les restaurants typiques éventuellement manger à l'extérieur en été. Elle arrive au centre-ville et il est l'heure du déjeuner, il est peu probable qu'il va pleuvoir! En utilisant son appareil personnel, elle demande des suggestions sur les endroits où manger.

Supposons que Dora a fait son choix et une fois là bas, elle découvre qu'elle ne peut pas manger à l'extérieur parce qu'il pleut. Par conséquent, Dora sera déçue et peut-être qu'elle dira à MyMap: *Hey, tu ne sais pas qu'il pleut! Pourquoi m'as-tu suggéré ce restaurant?*

Une des limites de MyMap est qu'il ne prenne pas en considération les situations exceptionnelles qui pourraient rendre inutile la suggestion et de réduire ainsi la confiance des utilisateurs envers le système. De Carolis et al [16] ont décidé de traiter ce genre de situations en étendant MyMap avec des règles filtrant l'ensemble de suggestions à recommander avant de proposer des lieux d'intérêt.

Dans la figure 20 des exemples de règles utilisées dans MyMap sont présentés. La première colonne décrit quel attribut est évalué pour l'application de la règle (il peut s'agir d'un attribut ou sa valeur), la deuxième colonne indique la catégorie à laquelle les éléments de la règle peuvent être appliquées (par exemple, monument, événement, ou tous ce qui est disponible), la troisième désigne le contexte de services de prestation qui doit être utilisé pour vérifier les fonctionnalités du contexte (par exemple, service météo) décrites dans la quatrième colonne (par exemple, s'il pleut). La cinquième colonne indique la stratégie de révision qui doit être utilisée (par exemple, utiliser d'autre préférences ou ne pas recommander cet article), la dernière colonne indique quelle stratégie communicative à appliquer pour interagir avec l'utilisateur (par exemple, indiquer les recommandations différemment sur la carte et expliquer pourquoi elles n'ont pas été faites).

Preference	Category	Checking Action	Context Condition	Revision Strategy	Explanation
Open Air	All	meteo(local)	Raining	Same category: -Keep matching other preferences -Not recommendable Open Air -Check explicit preferences	Indicate on the map differently Explain why
Open Air	All	meteo(local)	Sunny Temperature >30° Humidity= high Wind=low 10.30<Time< 17.30	Same category: - Keep matching other preferences -Not recommendable Open Air -Check explicit preferences	Indicate on the map differently Explain why
...	...	...	...	...	
-	Monument /Event	Check_if_open (local)	-	Same category: Exclude that item	Indicate on the map differently Explain why

Figure 20. Exemple de règles étendant MyMap

B. De Carolis, I. Mazzotta, N. Novielli and V. Silvestri [16] envisagent, en plus des règles, d'expliquer à l'utilisateur pourquoi certaines recommandations n'ont pas été suggérées en lui permettant de cliquer, par exemple, sur un triangle jaune pour obtenir les raisons pour lesquelles cet élément n'est pas recommandable ou non disponible. La figure 21 montre le triangle jaune sur lequel l'utilisateur peut cliquer pour avoir les explications.

Figure 21. Exemple de recommandation utilisant MyMap

8.1.6 Sourcetone Interactive Radio - Musiques selon l'humeur

En 2004, Malcolm Goodman et le Dr Jeff Berger ont fondé un système de recommandation qu'ils ont nommé Sourcetone radio interactive (www.sourcetone.com) [36], ce système demande à l'auditeur de préciser son état d'esprit ou émotionnel (le contexte) avant de lui recommander une chanson. La figure 22 montre une recommandation de la chanteuse « Lady Gaga » pour un utilisateur qui a un état d'esprit « agité ».

Figure 22. Imprime écran du site www.sourcetone.com

8.1.7 Amazon

Un autre système de recommandation sensible au contexte est celui d'Amazon [4] qui a été créé par Jeff Bezos en juillet 1995 et dont la filiale française a ouvert en 2000.
Amazon.com est une entreprise de commerce électronique américaine basée à Seattle. Sa spécialité la plus connue est la vente de livres, mais elle est diversifiée dans d'autres produits, notamment dans la vente de tout type de produits culturels : disques CD, musique en téléchargement, DVD, appareils photos numériques, matériels informatique et électroménager, etc. Ce système demande aux utilisateurs de se connecter via leur nom (voir figure 23.A) afin de créer leur profile et fournit un bouton « trouver un cadeau» (voir figure 23.B) pour chaque utilisateur afin de faire la distinction entre les préférences propres à l'utilisateur et les préférences de la personne à laquelle il va offrir ce cadeau.

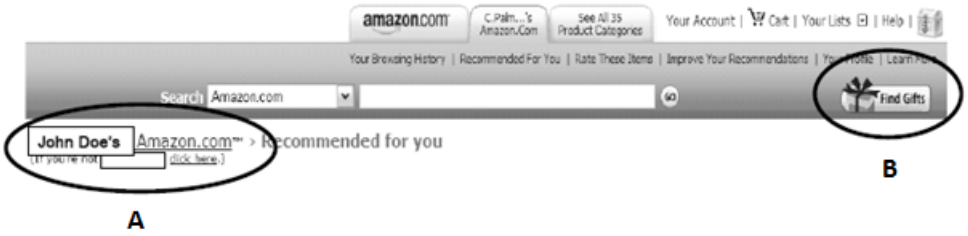

Figure 23. Barre d'outils du site www.amazon.com

8.1.8 Approche basée sur le Modèle d'Accès Personnalisé (PAM)

Cette approche consiste à étudier les possibilités d'intégration des informations contextuelles dans les systèmes de recommandation classiques en suivant les travaux initiés par G.Adomavicius et al. [2], Anand et al. [5], et L. Baltrunas. [7].

Le modèle d'accès personnalisé (PAM) [10] offre un ensemble de services, de concepts et de techniques de personnalisation, qui sont adaptables à différents types d'application. Cette plateforme satisfait les exigences suivantes :

- Dispose des métas modèles de profil et de contexte qui lui permettent d'être adaptable à toute sorte d'applications.
- Être indépendante.
- Fournir un ensemble de services qui peuvent personnaliser des applications existantes (utilisation partielle), ou même construire de nouvelles applications de personnalisation (utilisation complète).
- Rendre les informations réutilisables en assurant leur pérennité.

La figure 24 représente l'architecture de cette plateforme. Elle se décline en 3 couches: (i) une couche de persistance, (ii) une couche fonctionnelle et (iii) une couche de communication.

La couche de persistance: Concerne le stockage des informations contenues dans les métas modèles du profil et du contexte dans une base de données, assure leurs pérennité, et propose une API qui permet l'accès à ces informations.
La couche fonctionnelle: Regroupe l'ensemble des services offerts par la plateforme. Ces services sont : l'instanciation, la mise à jour, la contextualisation du profil, l'appariement des profils, le filtrage des profils par rapport au contexte et enfin, la reformulation des requêtes.
La couche de communication: N'est autre que l'interface graphique. Elle effectue la liaison entre le modèle d'accès personnalisé et l'utilisateur, et lui permet d'accéder à la base de données des profils et contextes, et d'exécuter les services proposés par la plateforme.

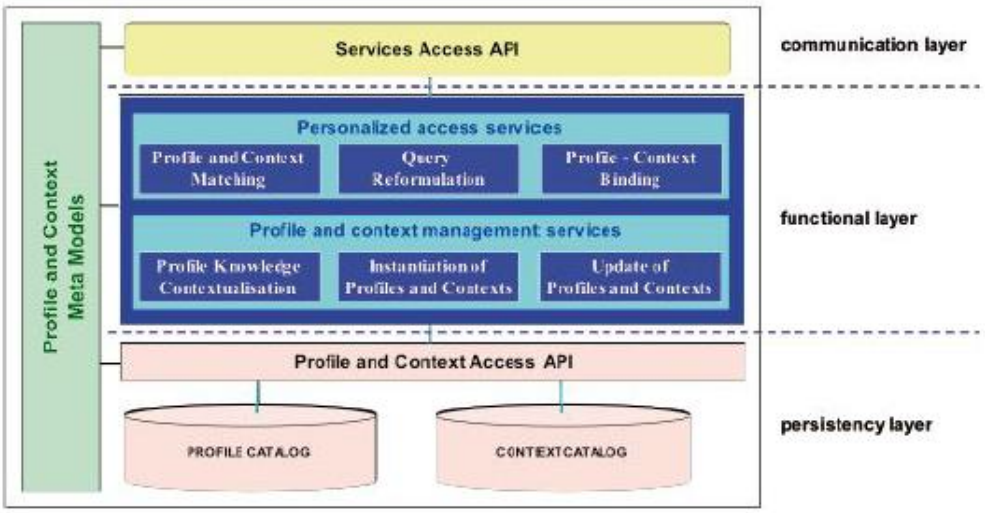

Figure 24. L'architecture du modèle d'accès personnalisé (PAM)

Toutes les composantes des trois couches du PAM, ont été implémentées conformément aux métas modèles, ce qui facilite leur adaptation à plusieurs applications. De plus, tous les messages adressés au PAM doivent respecter le format de base.

Le méta modèle du profil utilisateur contient les différentes informations qui décrivent l'utilisateur et ses préférences. Il existe cinq dimensions principales d'un profil utilisateur : données personnelles, centre d'intérêt, données de qualité, données de livraison et données de sécurité.

Le profil utilisateur ne suffit pas à lui seul pour décrire l'utilisateur et ses préférences de façon complète, par exemple, un utilisateur veut écouter des news le matin, et de la musique quand il conduit. L'introduction de la notion du contexte permet de pallier cette insuffisance d'informations.
S. Bouakaz et S. Abbar ont proposé un prototype de système de recommandation sensible aux contextes et ils ont spécifié les structures de profil et contexte utilisées, ainsi que les algorithmes de matching, en plus des algorithmes des autres services (contextualisation, binding).

Le prototype de système de recommandation sensible au contexte est basé sur des informations additionnelles afin de prédire le comportement de l'utilisateur. Le modèle d'accès personnalisé est instancié dans cette application réelle, où ils font appel à ses services, en les adaptant à leurs besoins.

Le service de contextualisation permet de détecter le changement de comportement de l'utilisateur en fonction de son contexte, et crée des mappings entre les préférences de l'utilisateur et les contextes dans lesquels la personnalisation doit être prise en considération.
A défaut de prendre tout le profil utilisateur en compte lors de la personnalisation, le service de binding filtre les informations inutiles qui n'appartiennent pas au contexte dans lequel l'utilisateur interagit.

Le système de recommandation PAM combine la méthode basée sur le contenu et le filtrage collaboratif. La méthode basée sur le contenu nous fournit les informations sur le profil utilisateur en analysant le contenu de ses préférences; ces profils sont par la suite comparés afin de déterminer les utilisateurs qui sont similaires et effectuer des recommandations collaboratives. Quant au filtrage collaboratif, il exploite les votes donnés par les k voisins similaires à un utilisateur actif afin de déduire les votes manquants en utilisant des fonctions d'agrégation.

L'architecture de ce système de recommandation sensible au contexte : CARS (Figure 25) est composée de deux parties : La partie en haut à gauche concerne l'acquisition des informations, et la partie en bas à droite concerne la personnalisation des recommandations.
Le processus d'acquisition de l'information : Acquiert et gère les informations dont le CARS a besoin pour effectuer les recommandations. Ces informations sont regroupées en trois entités: Profils utilisateur, les descriptions des contenus et les contextes.
Les recommandations personnalisées : Ce bloc calcule les recommandations contextuelles. Le système de recommandation prend en entrée les profils contextuels générés du bloc d'acquisition de l'information et le contexte courant, et donne en sortie la liste des contenus à recommander. Lors de l'évaluation des contenus recommandés, le comportement de l'utilisateur actif est enregistré dans ses logs, sur lesquels le bloc d'acquisition de l'information se base pour mettre à jour le profil utilisateur, le contexte et les mappings résultant de la contextualisation.

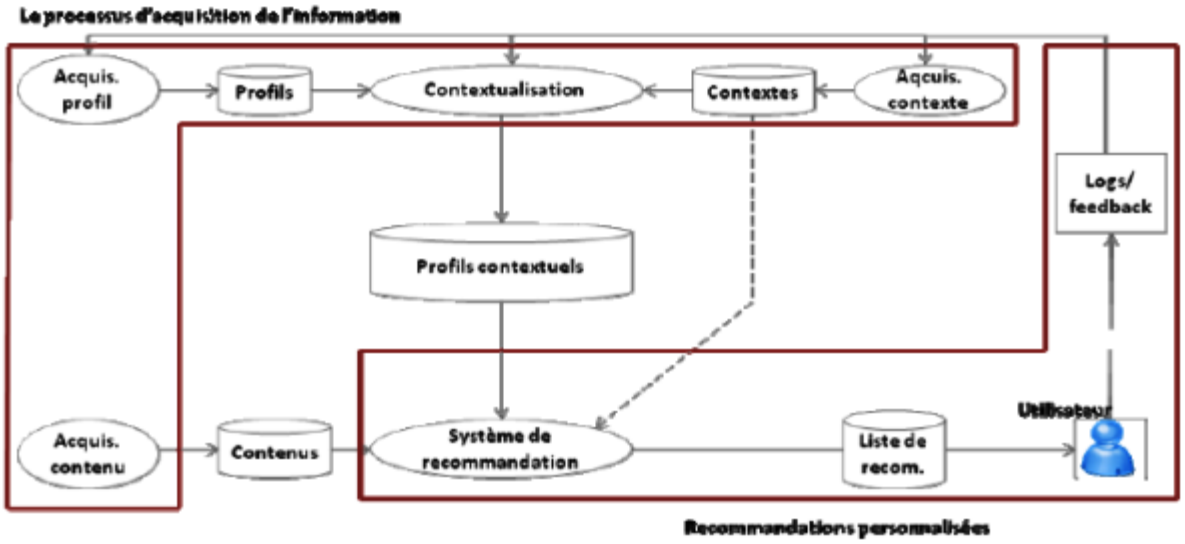

Figure 25. Architecture globale du système de recommandation sensible au contexte

Ce système de recommandation sensible au contexte répond aux exigences suivantes:

- Faire une distinction entre le profil utilisateur et le contexte pour bien comprendre chaque entité.
- Permet la découverte du contexte des fichiers journaux.
- Prend en considération le contexte, en plus du profil utilisateur, lors des recommandations et part du fait que les préférences de l'utilisateur changent selon son contexte.

8.2 Comparaison

Le tableau 7 montre une comparaison entre les différentes approches étudiées dans cette thèse selon le type d'approche de recommandation utilisé, le type de la collecte des données, le type du contexte et sa définition, ainsi que le domaine d'évaluation, et la flexibilité de l'approche.

8.2.1 Approche de recommandation

Il existe trois types de systèmes de recommandation : Les systèmes basés sur le filtrage à base de contenu (FBC), les systèmes basés sur le filtrage collaboratif (FC) et les systèmes hybrides. Dans les systèmes FBC, les descripteurs de contenus sont directement appariés avec les profils utilisateurs pour estimer leurs utilités. Seuls les contenus ayant un score d'utilité important seront recommandés aux utilisateurs. Les systèmes FC sont basés sur l'observation selon laquelle les utilisateurs ayant le même comportement (passé) ont probablement les mêmes centres d'intérêt. Ces systèmes n'ont besoin d'aucune description de contenu. Leurs idée principale est de recommander, à un utilisateur, les contenus consommés (appréciés) par les utilisateurs qui lui ressemblent. Les systèmes hybrides, enfin, combinent les techniques du filtrage basé sur le contenu et celles du filtrage collaboratif pour améliorer la pertinence des recommandations. Un exemple de ces systèmes est la collaboration par le contenu dans lequel la technique FBC est utilisée pour apprendre les profils des utilisateurs. Ces profils sont ensuite utilisés pour calculer des recommandations en utilisant la technique du FC.

8.2.2 Type de la collecte de données

La collecte de données peut être explicite ou implicite. Elle est explicite quand l'utilisateur introduit ses préférences d'une manière manuelle (cas de système Sourcetone), ou implicite lorsque le système déduit le profil de l'utilisateur selon le comportement de se dernier (cas de système Discover). La plupart de ces systèmes ont eu recourt à la collecte implicite de données, qui ne nécessite pas un effort de la part de l'utilisateur, même s'il est assez difficile d'obtenir ces données.

8.2.3 Type du contexte

Le Pre-Filtering est la technique la plus utilisée car elle est beaucoup plus simple à appliquer et moins coûteuse que la méthode du Post-Filtering. Le système MyMap par exemple utilise la technique Pre-Filtering contrairement au système DaVI qui opte pour la technique de Post-Filtering.

8.2.4 Définition du contexte

Certains systèmes ont défini le contexte en se basant sur le facteur temps comme dans l'approche Micro-profiling. D'autres systèmes ont utilisé l'état d'esprit (l'humeur) de la personne comme information contextuelle pour recommander de la musique (cas du système Sourcetone). Le contexte, pour d'autres systèmes, peut être n'importe quelle entité, selon les besoins de l'utilisateur. C'est le cas de l'approche MyMap où le contexte n'est pas défini (spécifié par « quelconque » dans le tableau 7).

8.2.5 Domaine d'évaluation

Il précise le domaine dans lequel les systèmes ont été évalués. Le système News@hand est évalué dans le domaine des articles d'informations par ailleurs le système DaVi est évalué dans n'importe quel domaine « quelconque ».

8.2.6 Flexibilité

Ce critère indique si le système peut être utilisé dans plusieurs domaines (plus les domaines sont nombreux, mieux c'est). Par exemple l'utilisation du système micro-profiling n'est pas flexible par contre celle du système Discover est flexible.

Tableau 7. Tableau récapitulatif des différents Systèmes de recommandation sensibles aux contextes

Critères \ Approches	Micro-Profiling	DaVI	Discover	News@hand	MyMap	Sourcetone	Amazon	PAM
Approche de recommandation	CF	CF+AR	Hybride	Hybride	CBF	-	CF+CBF	CF+CBF
Type de la collecte des données	Implicite	Implicite	Implicite	Implicite	Implicite / Explicite	Explicite	Implicite / Explicite	Implicite / Explicite
Type du contexte	Pre-filtering	Post-filtering	Pre-filtering	Pre-filtering	Pre-filtering	Pre-filtering	Pre-filtering	Pre-filtering
Définition du contexte	Temps	Accès à un item ou page web	Quelconque	Construit en temps réel	Quelconque	Etat d'esprit	Quelconque	Quelconque
Domaine d'évaluation	Musique	Quelconque	Quelconque	Article d'information	Tourisme	musique	e-commerce	Films
Flexibilité	Non	Oui	Oui	Non	Non	Non	Non	Oui

9. Conclusion

Dans ce chapitre, nous avons défini le contexte, ainsi que son importance, ses dimensions, et la pertinence de son choix. Ensuite nous avons présenté les différentes méthodes de son introduction, ainsi que les différents travaux sur les systèmes de recommandation sensibles aux contextes. Enfin, une comparaison entre les différents systèmes existants a été introduite sur la base d'un certain nombre de critères définis.

Le prochain chapitre sera consacré à la partie conception et réalisation de notre système de recommandation sensible au contexte.

CHAPITRE III

Conception et Réalisation

1. Introduction

La prise en compte du contexte par les systèmes de recommandation permet une meilleure modélisation du comportement des utilisateurs, en fournissant des informations supplémentaires qui peuvent être exploitées afin d'effectuer de meilleures recommandations.
Ce chapitre est consacré à la partie conception et réalisation de notre application ([37], [38]), qui consiste à recommander des items (comme par exemple, des films, livres, chansons, restaurants, etc.) à un utilisateur donné en tenant compte du contexte de la recommandation. Dans notre étude, nous avons introduit le contexte intrinsèque représenté par un sous ensemble d'attributs décrivant l'item lui-même, par ailleurs le contexte extrinsèque qui est défini sur l'ensemble d'attributs relatifs à l'environnement de l'utilisateur à été donné comme perspectif. Nous présentons, également, l'architecture générale de l'application avec ses différents modules, un exemple illustrant la méthode proposée, l'ontologie utilisée ainsi que l'environnement de travail choisi.

2. Modèle de données

Dans ce qui suit, nous donnons l'ensemble des définitions relatives aux concepts de base que nous manipulons dans ce chapitre.

Préférence :
Une préférence est une formule qui permet d'hiérarchiser un ensemble d'objets par rapport aux intérêts et aux besoins d'un utilisateur. Il existe deux types de préférences : qualitatives et quantitatives. Les préférences qualitatives sont exprimées avec des relations d'ordre binaire (comme par exemple, la relation '>') et permettent de définir un ordre partiel sur les objets. Les préférences quantitatives sont exprimées à travers des fonctions de scores qui attribuent des scores aux différents objets. Ce type de préférences permet de définir un ordre total sur les objets. Dans le reste de ce chapitre, nous nous focaliserons sur les préférences quantitatives que nous modélisons comme suit : Une préférence *pi* est un couple *pi(pri,wi)* où *pri* est un prédicat de type <attribut, opérateur, valeur> et *wi* est un nombre réel compris entre 0 et 1 qui représente le degré d'intérêt de l'utilisateur par rapport au prédicat *pri*. La valeur 0 reflète la préférence minimale alors que la valeur 1 reflète la préférence maximale.

Exemple : Soit R un schéma de relation modélisant des voitures, R (marque, type, prix, couleur, kilométrage, puissance, nbChevaux, nbPlaces). Un utilisateur peut définir les préférences suivantes : *p1* (<couleur,=,'Rouge'>, 0.9) et *p2* (<Prix,'>', 1.4M-DA>, 0.2).
La préférence *p1*, exprime le fait que l'utilisateur accorde un score de 0.9 aux tuples de la relation R (voiture) pour lesquels l'attribut couleur prend la valeur rouge. En même temps, cet utilisateur accorde un score de 0.2 aux tuples pour lesquels l'attribut prix prend une valeur supérieur à 1.4 million de Dinars Algériens.

Contexte :
Dans ce mémoire, nous définissons le contexte du point de vu des objets et non pas de celui des utilisateurs. Le contexte est toute information qui influe sur l'intérêt que peut exprimer un utilisateur sur un objet. Nous modélisons le contexte comme une conjonction de prédicats définis sur des attributs contextuels. Nous distinguons deux types de contextes : contexte **intrinsèque** et contexte **extrinsèque**. Le contexte intrinsèque est défini par un sous ensemble des attributs décrivant l'objet lui-même.

Exemple : Dans la relation R, les attributs kilométrage et type peuvent être considérés comme des attributs contextuels intrinsèques.
Le contexte extrinsèque est un ensemble d'attributs relatifs à l'environnement d'interaction de l'utilisateur. Il peut renseigner sur l'emplacement géographique de l'utilisateur (adresse IP), sur la période temporelle de son interaction (date et heure), sur le media utilisé pour l'interaction, etc.

Exemple : A partir de la date et de la localisation par exemple, on peut détecter un évènement qui peut changer les préférences de l'utilisateur de façon momentanée. Ainsi, un utilisateur Algérien peut se voir recommander des objets historiques et révolutionnaires (Livres, chansons, films, documentaires, etc.) dans les périodes précédant et succédant un évènement historique comme le 1er Novembre ou le 5 juillet.

Préférence contextuelle :
Afin de capturer la variabilité incontestable des préférences d'un utilisateur en fonction des contextes, nous avons introduit le concept de préférence contextuelle. La préférence contextuelle est une préférence valide dans un contexte particulier. Elle ne doit être prise en compte que lorsque l'utilisateur se trouve dans ce contexte. Nous utilisons le formalisme suivant pour modéliser la préférence contextuelle : *pc* : <*pi* | contexte=*cj*>. Où *pi* est une préférence quantitative normale valide dans le contexte *cj*.

Exemple : toujours sur la relation des voitures R, un utilisateur peut définir les deux préférences contextuelles suivantes : *cp1* : [(<Prix,'>', 1.4M-DA>, 0.2) | type='Renault'], *cp2*:[(<Prix,'<', 1.7M-DA>, 0.8) | type='BMW']. Dans *cp1*, l'utilisateur exprime le fait qu'il accorde un score de 0.2 aux tuples pour lesquels l'attribut prix prend une valeur supérieur à 1.4 million de Dinars Algériens, dans le contexte des voiture de type Renault (contexte intrinsèque). En même temps, l'utilisateur accorde un score de 0.8 aux voitures ayant un prix pouvant atteindre 1.7 million DA dans le contexte des voitures de type BMW.

Profil utilisateur :
Dans notre approche, le profil utilisateur est défini par l'ensemble des préférences contextuelles implicites (déduits automatiquement par le système en analysant les logs d'interactions de l'utilisateur). *Pu* = {*cp1*, …, *cpn*}. Cette modélisation permet de capturer le fait qu'un utilisateur puisse avoir différents scores de préférence sur le même objet en fonction de son contexte courant.

Exemple : un utilisateur peut accorder un score très élevé au genre 'films de guerre' lorsque le contexte est relatif à un évènement historique, et en même temps associer un score très faible au même genre dans les autres contextes de l'année.

Descriptif de contenu :
Au-delà des tuples d'une base de données, nous nous intéressons à la recommandation de contenus en général (livre, Url, article, produit, film, chanson, etc.). Dans ce chapitre, nous considérons que tous les contenus sont modélisés par des conjonctions de prédicats.
$Dc=\{pr_1, \dots pr_n\}$.

Exemple : *Dc* (livre1)={<titre,=, Le Chao des Sens>, <auteur,=, AhlamMosteghanemi>, <Editeur=France Meyer>, …}

Nous donnons dans la figure 26 le modèle Entité-Association relatif à notre application.

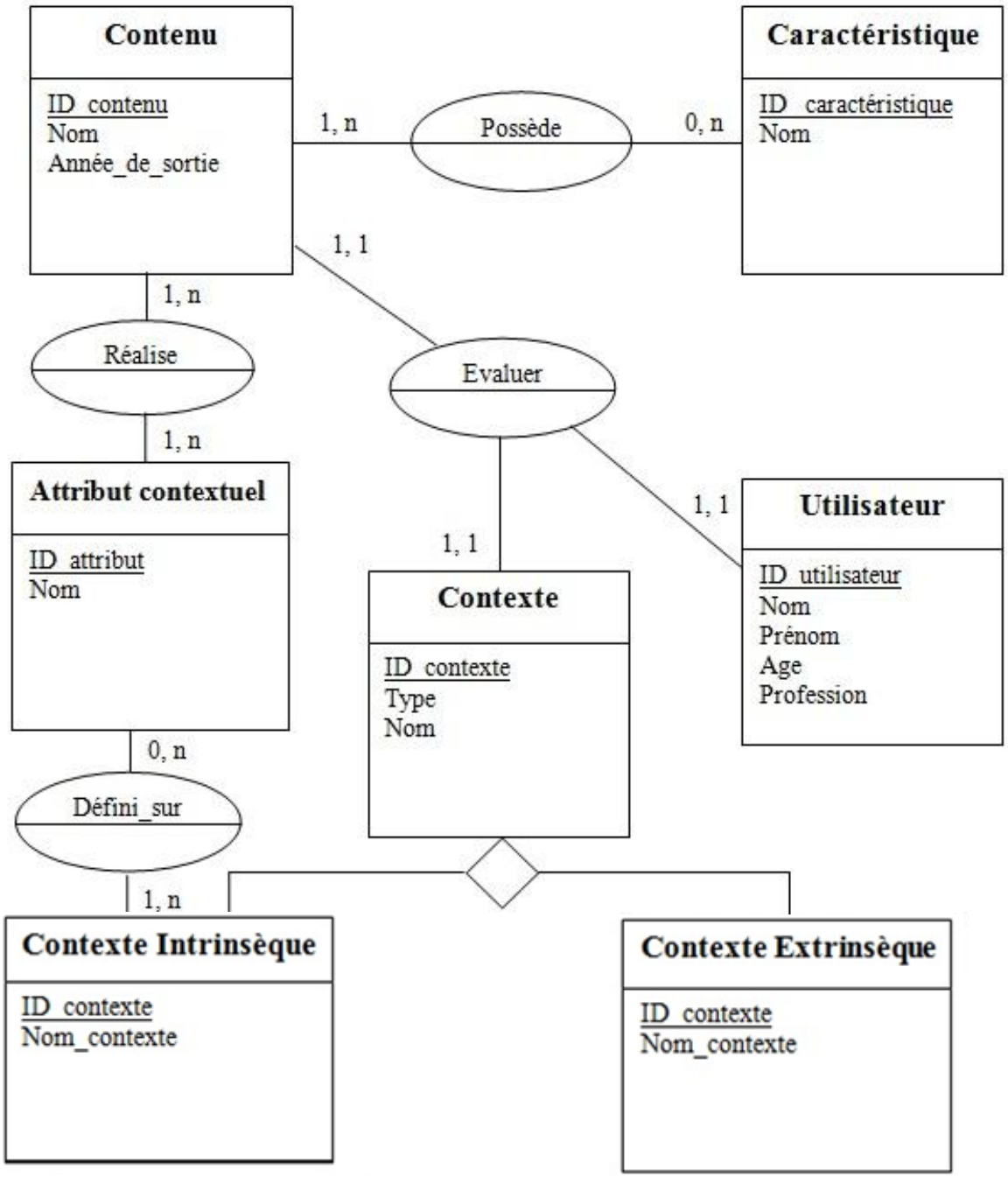

Figure 26. Modèle Entité-Association

La section suivante est dédiée aux diagrammes UML correspondants à notre application.

3. Conception de l'application

Pour décrire les différentes étapes de conception de notre application, nous avons opté pour le langage UML.

3.1 Diagramme de classe

Le diagramme de classe peut être facilement déduit de notre schéma entité association présenté dans la figure 26. Cette déduction se fait par l'application des règles de transformation suivantes :

- entité → classe.
- association avec attributs → classe associative.
- inversement des cardinalités.

3.2 Diagramme de Séquences

Le diagramme de séquences représente l'une des vues dynamiques les plus importantes d'UML. La figure 27 montre les interactions entre les différents objets de notre application.

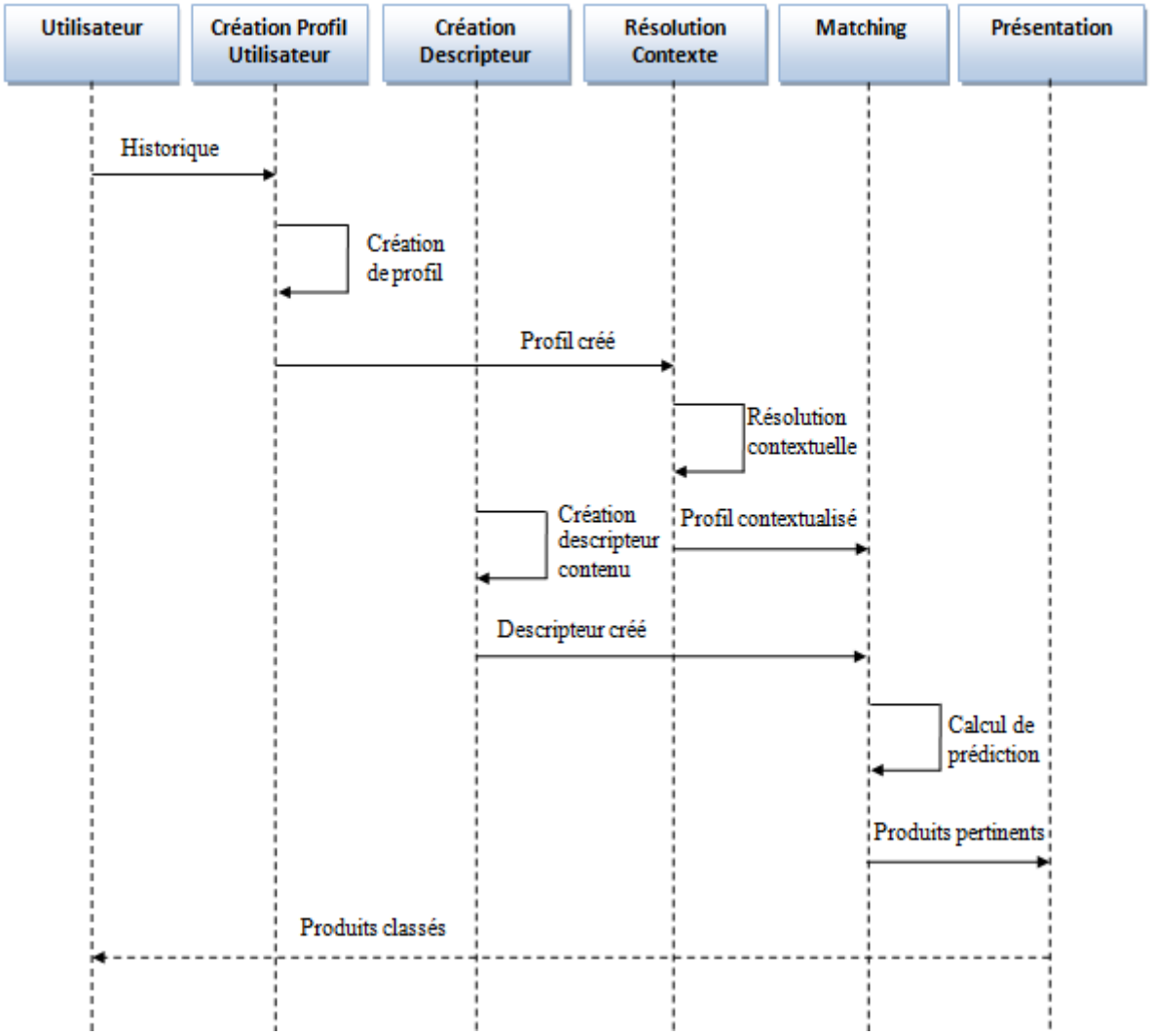

Figure 27. Diagramme de séquences

3.3 Diagramme de collaboration

Un diagramme de collaboration montre les relations entre les objets jouant les différents rôles. Toutefois, ce diagramme ne contient aucune notion de temps.

Après que l'utilisateur ait évalué un certain nombre de produits, le module « Création Profil Utilisateur » construit son profil à partir de son historique et l'envoi au module «Résolution Contexte », ce dernier fourni des profils contextualisés au module « Matching ». Le module « Matching » reçoit ainsi les descripteurs de contenus créés par le module « Création Descripteur Contenu » est calcule la similarité entre le profil utilisateur contextualisé et les différents descripteurs de contenus, pour ainsi renvoyer les contenus pertinents au module « Présentation ». Le module « Présentation » classe les contenus par ordre décroissant selon leurs degrés de similarité et les affiche pour l'utilisateur. La figure 28 illustre l'échange d'information entre les différents modules.

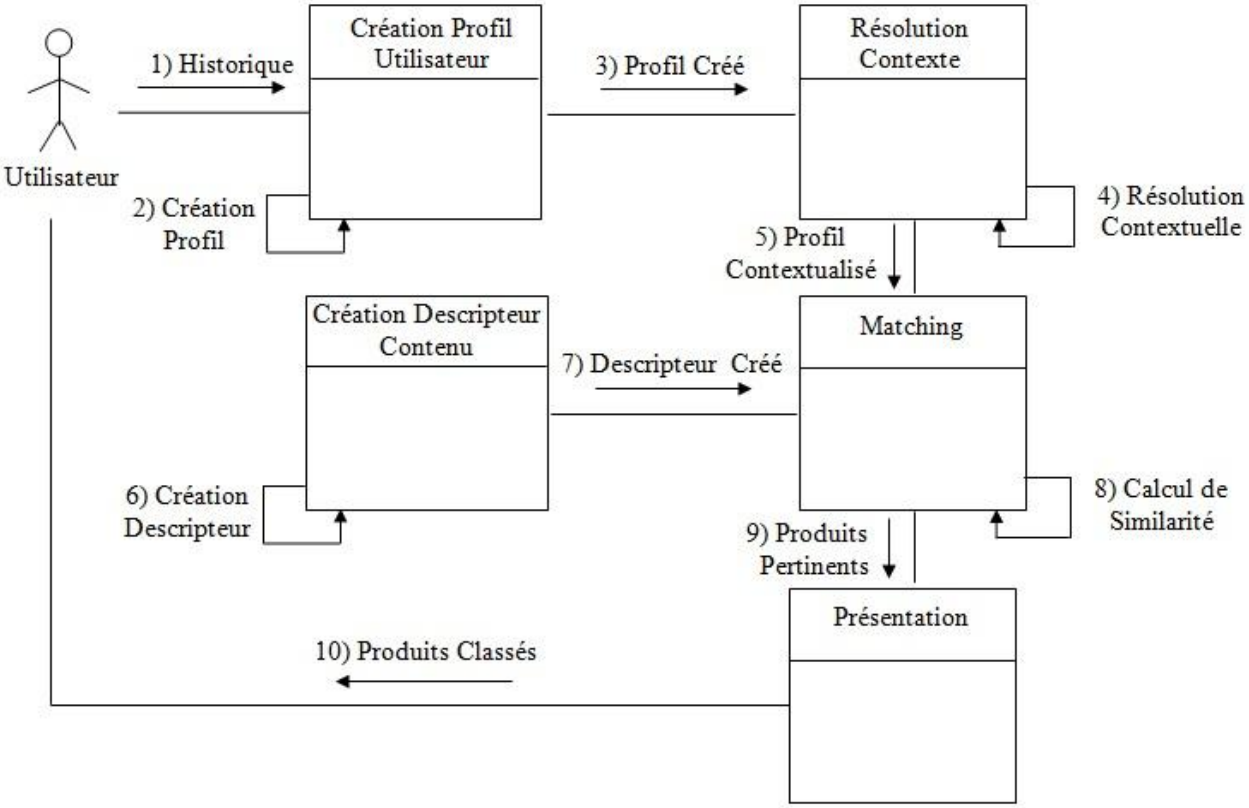

Figure 28. Diagramme de collaboration

Nous présentons dans la section suivante l'architecture générale de notre système de recommandation. Nous détaillerons par la suite les différents modules le composant.

4. Architecture générale de l'application

Nous avons opté pour l'utilisation de la méthode de *pré-filtrage* (définie au chapitre 2) pour l'introduction du contexte dans notre système de recommandation (figure 29). Dans cette approche, l'information contextuelle est utilisée afin de filtrer les évaluations qui ne correspondent pas au contexte courant de l'utilisateur. Le pré-filtrage est exécuté avant que le module principal de la recommandation ne soit lancé.

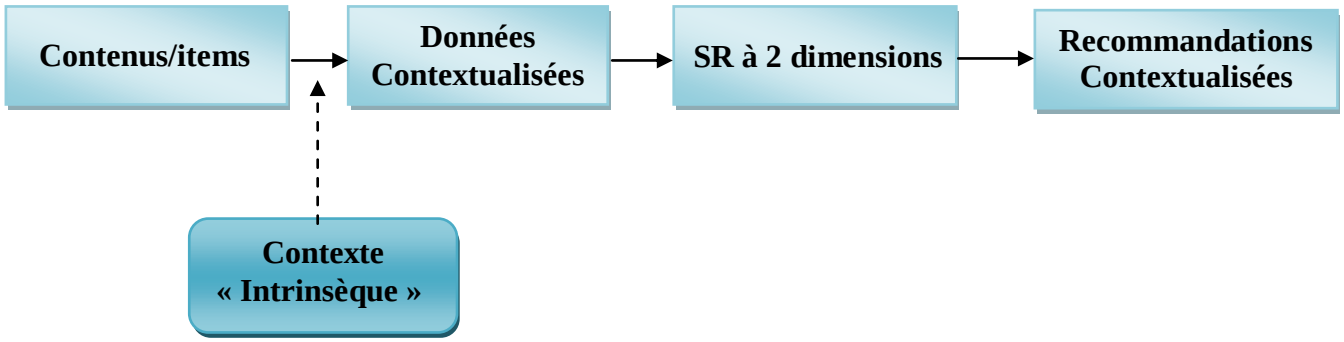

Figure 29. Méthode d'introduction du contexte

Comme le montre la figure 30, l'architecture globale de notre système de recommandation comporte six modules principaux. Cette architecture vient compléter celle présentée dans [6] qui ne considérait que la recommandation basée sur le FBC sans la prise en compte du contexte.

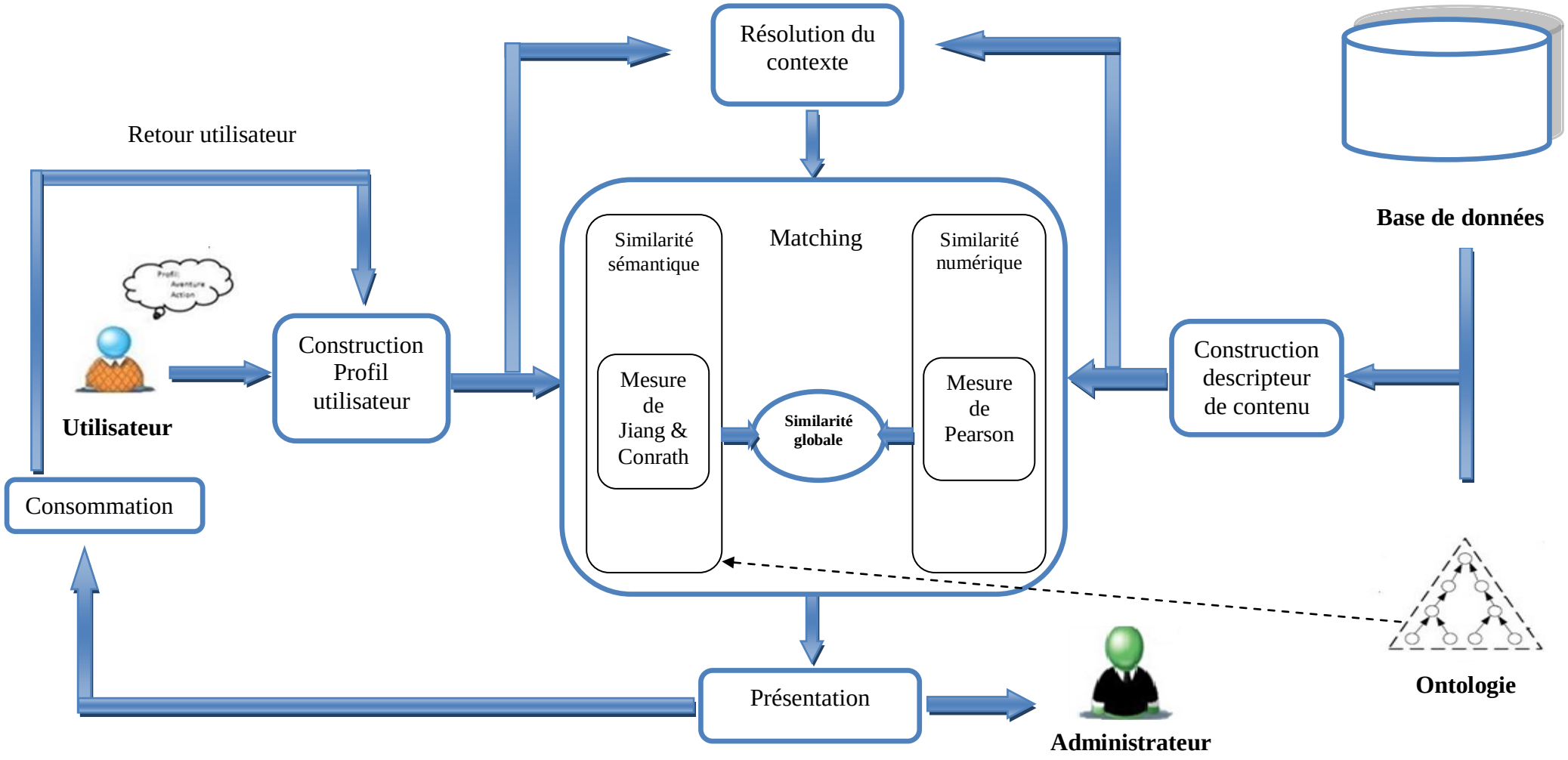

Figure 30. Architecture générale de l'application

1. **Création descripteur de contenu :** ce module a pour tâche de créer un descriptif produit à partir des informations stockées dans la base de données et l'ontologie.

2. **Création profil utilisateur :** ce module a pour tâche la création des profils utilisateurs. Cela correspond, dans notre application, au calcul de préférence pour chaque utilisateur en se basant sur le contexte intrinsèque.

3. **Matching :** ce module sert à calculer les recommandations pertinentes. Nous avons utilisé trois types de similarité :

   Similarité sémantique : nous avons implémenté la mesure de Jiang & Conrath dont l'équation a été présentée dans la section 5.3 du chapitre 1.

   Similarité numérique : la mesure de Pearson a été utilisée pour calculer le taux de corrélation entre le profil utilisateur et le produit**.**

   Similarité globale : notée $Sim_{globale}$, elle combine les deux similarités : numérique ($sim_{préf}$) et sémantique ($sim_{sém}$) selon la formule suivante:

   $$Sim_{globale}(Pu, C) = \alpha \times sim_{sém}(Pu, C) + \beta \times sim_{préf}(\overrightarrow{Pu}, \overrightarrow{C})$$

   Avec $(\alpha + \beta = 1)$. Nos expérimentations répétées nous ont conduits à choisir les valeurs 0.75 pour α et 0.25 pour β.
   Cette mesure a été présentée et détaillée dans [6].

4. **Résolution du contexte :** ce module à pour tâche la contextualisation du profil utilisateur afin de lui recommander des produits plus pertinent en utilisant les deux algorithmes cités présenté en détail dans la section 5 de ce chapitre.

5. **Présentation :** après l'obtention des calculs effectués par le module « Matching », ce module se charge de :

   - Classer les produits par ordre décroissant du score de similarité qu'ils ont obtenu et de les afficher à l'utilisateur du plus important au moins important.

   - Afficher les courbes d'évaluation pour l'administrateur de notre application.

6. **Consommation :** dans ce module, l'utilisateur parcourt la liste des recommandations et choisit les contenus qui lui sont proposés. Les produits consommés et notés par l'utilisateur sont stockés dans des logs de journalisation afin d'enrichir son profil ultérieurement.

Dans ce qui suit, nous donnons les définitions relatives aux concepts principaux de notre approche de recommandation contextuelle.

5. Recommandation contextuelle

Dans cette section, nous allons détailler les trois modules clés de notre système [38], à savoir : *Création de profil, résolution du contexte et appariement profil/contenu.*

5.1 Création des profils utilisateurs

Le concept de profil utilisateur est la pierre angulaire de tout système de recommandation. Nous proposons dans ce mémoire une nouvelle approche pour la création automatique des profils basée sur l'analyse des logs d'interaction (appelés aussi fichiers de journalisation). Nous rappelons qu'un profil utilisateur est un ensemble de préférences contextuelles. Nous expliquerons, dans ce qui suit, comment créer des profils à partir des logs d'interaction en considérant des contextes intrinsèques.

Algorithme 1 : Création des profils

1. **Entées** :
2. L : log de journalisation, C : ensemble de contextes, T : ensemble de tuples
3. **Sortie** :
4. P_u : Profil utilisateur
5. **Début**
6. C := Instances_Contextes(L)
7. T := Prédicats(L)
8. Pu := {}
9. **Pour** chaque c_i dans C **faire**
10. **Pour** chaque t_j dans T **faire**
11. s_{ij} := Score(t_j, c_i, L)
12. P_u=P_u U {(t_j, s_{ij}| c_i)}
13. Fin Pour
14. Fin Pour
15. **retourner** P_u.
16. -**Instance_Contextes(L)** : retourne l'ensemble de toutes les instances du contexte
17. intrinsèque présentés dans les fichiers de journalisations L.
18. -**Predicats(L) :** retourne l'ensemble de tous les prédicats non contextuels dans L
19. pertinent pour le profil utilisateur.
20. -**Score(t, c, L) :** retourne la moyenne des scores assignés au prédicat t dans le contexte c.
21. **Fin**

Les logs d'interaction que nous utilisons ont une structure commune à plusieurs systèmes de recommandation commerciaux (ex. MovieLens, LastFm, Pandora, etc.). Nous nous intéressons dans ces logs aux votes laissés /donnés par les utilisateurs sous la forme : Vote <$content_l$, $utilisateur_k$, $score_{lk}$>, où l'utilisateur (k) accorde un score de préférence ($score_{lk}$) au contenu (l). La création des profils se fait en deux étapes :

1. Préparation des données : afin de pouvoir extraire des préférences contextuelles à partir des logs, nous transformons ces derniers par expansion des votes et propagation des scores en se basant sur les descriptifs des contenus. Nous obtenons après transformation des tuples avec la forme suivante :<$utilisateur_k$, $contexte_i$, $predicat_j$, $score_{lk}$>, où contexte_i est un prédicat contextuel (contexte intrinsèque) décrivant le contenu_l, pedicat_j est un prédicat non contextuel du contenu_l et score_lk n'est autre que le score exprimé par l'utilisateur_k sur le contenu_l.

2. Calcul des profils : après l'étape de la préparation des données, nous calculons la moyenne des $scores_{lk}$ exprimés par l'utilisateur pour chaque $predicat_j$ dans chaque $contexte_i$.

Algorithme 1 décrit le processus de création des profils à partir des logs transformés. L'idée principale de cet algorithme est de calculer le score contextuel de chaque prédicat dans chaque instance de contexte. Ceci permettra à notre système de capturer la variabilité des préférences en fonction des contextes. En d'autres termes, ce processus permet de modéliser le fait qu'un utilisateur ait différents niveaux de préférences sur un même objet.

Exemple : soit le profil suivant : Samira (<genre_chanson=sentimentale,0.9|chanteur=Céline Dion>, <genre_chanson= sentimentale,0.1 |chanteur=Lady Gaga>). Dans ce profil Samira exprime le fait qu'elle donne un score élevé aux chansons sentimentales lorsque celles-ci sont interprétées par Céline Dion est un score très faible aux mêmes genres de chanson lorsque celles-ci sont interprétées par Lady Gaga.

5.2 Résolution du contexte

Le second composant important de notre Système de Recommandation (SR) est celui de résolution des contextes. Dans le cas général, ce composant prend en entrée les descripteurs des contenus candidats à la recommandation, le profil utilisateur ainsi que des informations relatives à la situation de l'utilisateur (ex. adresse IP, date et heure de connexion, média utilisé, etc.) et retourne les contextes intrinsèque et extrinsèque à considérer dans le calcul des recommandations faites à l'utilisateur. Nous décrivons dans ce qui suit le processus de résolution des contextes intrinsèques.
Dans son schéma classique, le processus de résolution de contexte procède à l'extraction du contexte intrinsèque directement à partir des contenus identifiés comme candidats potentiels à la recommandation. Ceci revient à retourner tous les prédicats contextuels décrivant les contenus candidats. Mais dans certains cas, les contextes extraits par ce processus ne sont pas compatibles avec le profil de l'utilisateur.

Exemple : Dans le cas du profil de Samira par exemple, nous pouvons distinguer deux contextes (chanteur = Céline Dion) et (chanteur = Lady Gaga). Si l'on suppose qu'une chanson sentimentale de Shakira est identifiée comme candidate à la recommandation, alors le processus de résolution de contexte va retourner le contexte (chanteur = Shakira). Cependant, le système de recommandation ne saura pas quelle préférence de Samira prendre en compte pour évaluer la pertinence de cette chanson (0.9 ou 0.1 ?).

Algorithme 2 : Résolution des contextes

1. **Entées** :
2. C := Instances_Contextes(ensembleCandidats) // instances de contexte contenues
3. dans les contenus candidats à la recommandation.
4. C_u := Instances_Contextes(Pu) //instances de contextes contenues dans le profil Pu
5. **Sortie** :
6. C_c : ensemble de pairs(ci,xi) où c_i est un contexte courant. x_i est un facteur de
7. correction des préférences utilisateurs, x_i dans [0,1].
8. **Début**
9. C_c := {}
10. **Pour** chaque c_i dans C **faire**
11. **Si** c_i dans C_u **alors**
12. C_c:= C_c U {(c_i,1)} // pas de correction
13. **Sinon**
14. c^* := {c^*|sim(c^*,c_i)=Max(sim(c^*,c_j)), j=1..||C_u||}
15. C_c:= C_c U {(c^*,sim(c^*,c_i))}
16. **Fin Si**
17. **Fin Pour**
18. **retourner** C_c.
19. **-sim(c_1,c_2)** : retourne la similarité entre les deux instances de contexte c_1 et c_2.
20. **Fin**

Pour résoudre ce problème, nous proposons une approche basée sur la similarité entre les instances du contexte. L'intuition de notre approche est que les utilisateurs aiment pareillement les produits similaires. Lorsqu'un nouveau contexte se présente à l'utilisateur, nous procédons au calcul de similarité entre le nouveau contexte et l'ensemble de tous les contextes présents dans le profil de l'utilisateur. Il conviendra ainsi de retenir le contexte existant (dans le profil) qui est le plus similaire au nouveau contexte qui se présente.

Exemple : Si Samira aime les chansons interprétées par des chanteurs similaires, et que nous avons l'information selon laquelle Sharika est plus similaire (chante le même style que) à Lady Gaga qu'elle ne l'est avec Céline Dion, alors il serait plus pertinent de considérer les préférences de Samira relatives à Lady Gaga (<genre_chanson= sentimentale,0.1 |chanteur=Lady Gaga>) pour évaluer les chansons de Shakira. Et comme Shakira n'est pas Lady Gaga, alors nous introduisons aussi un facteur de correction des préférences qui est donné par la similarité entre Shakira et Lady Gaga.

Algorithme 2 décrit le processus de résolution des contextes. L'algorithme prend en entrée l'ensemble des instances de contexte données dans les contenus candidats ainsi que l'ensemble des contextes contenus dans le profil utilisateur, il renvoie un ensemble de paires (contexte, facteur de correction) compatibles avec le profil utilisateur.

5.3 Appariement profil/contenu

Le processus de base dans un SR est celui chargé d'évaluer la pertinence d'un contenu pour un utilisateur particulier. Certains SR décident qu'un contenu est pertinent pour un utilisateur si les utilisateurs qui lui ressemblent ont consommé ce contenu et l'ont considéré comme pertinent pour eux. On parle alors de SR basé sur le filtrage collaboratif (CF). D'autres SR prédisent la pertinence d'un contenu pour un utilisateur en calculant la similarité (corrélation) entre le profil utilisateur et le descripteur de contenu. On parle alors de SR basé sur le filtrage

à base de contenu (CBF). Dans ce qui suit, nous donnons une description de notre processus d'appariement entre profils et contenu dans le cas des systèmes de recommandation CBF.
Comme montré dans la figure 30, le processus d'appariement prend en entrée le profil utilisateur, un ensemble de contenus candidats ainsi que l'ensemble des contextes courants. Il renvoie une liste ordonnée de recommandations. Pour produire cette liste, le processus d'appariement calcule la similarité entre le profil utilisateur et chaque descripteur de contenu candidat. Ces derniers sont alors ordonnés dans l'ordre décroissant de leur pertinence à l'utilisateur (similarité avec le profil). Les K premiers contenus sont recommandés à l'utilisateur.

L'appariement entre un descripteur de contenu (DescCt) et un profil utilisateur (Pu) dans un contexte donné (Cxt), *Sim*(Pu, DescCt, Cxt), se fait en deux étapes :

1- Filtrage et correction des préférences pertinentes.
Cette étape consiste à choisir parmi toutes les préférences de l'utilisateur celles qui sont pertinentes au contenu (DescCt) dans le contexte (Ctx). Ces préférences sont ensuite corrigées en les multipliant par le facteur de correction correspondant au contexte (Ctx). Le résultat de cette étape est un vecteur de prédicats pondérés (tiré de DescCt ∩ Pu).
2- Evaluation de la pertinence du contenu.
Cette étape consiste à mesurer la similarité entre le profil issu de la première étape avec le descripteur de contenu. Nous avons utilisé la similarité qui combine la similarité sémantique et la similarité numérique comme illustré dans la section suivante.

5.4. Mesures de similarités utilisées dans l'application

Dans notre application, nous avons opté pour la mesure de Pearson pour la similarité numérique et la mesure de Jiang & Conrath pour la similarité sémantique.

Similarité de Pearson

Nous avons utilisé la corrélation de Pearson entre les deux vecteurs : utilisateur représenté par « $X(x_1, x_2,\ldots x_n)$ » et produit représenté par « $Y(y_1, y_2,\ldots y_n)$ »). La formule de Pearson est donnée par :

$$\text{Sim (x,y)} = r_p = \frac{\sum_{i=1}^{N}(x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

Mesure de Jiang et Conrath

La similarité de Jiang et Conrath est une mesure hybride qui combine une mesure structurelle (longueur du plus court chemin entre deux nœuds) et une mesure sémantique basée sur le contenu informationnel des nœuds comparés. La similarité de Jiang et Conrath s'utilise dans le cas des ontologies, et est définie par :

$$\boldsymbol{Sim\ (c1, c2) = \frac{1}{distance\ (c1, c2)}}$$

Où la distance entre *C1* et *C2* est calculée par la formule suivante :

Distance (c1, c2) = CI (c1) + CI (c2) – (2.CI (PPG (c1, c2)))

Avec :

CI = - Log (Nombre d'instances du concept / Nombre total d'instances)
PPG : Plus Petit Généralisant.

Nous combinons ensuite ces deux mesures, comme mentionné ci-dessus (dans la section 4), pour utiliser la similarité globale $Sim_{globale}$ afin d'améliorer les résultats de la pertinence des produits pour l'utilisateur actif (c'est-à-dire l'utilisateur demandeur de recommandations).

6. Exemple illustratif pour la recommandation des films

Nous allons donner dans cette section un exemple de calcul de recommandation sur des films réalisés par un réalisateur connu en premier lieu et un réalisateur inconnu en second lieu pour nos utilisateurs. Nous avons pris comme exemple l'utilisateur 424 avec son profil montré dans le tableau 8.

Tableau 8. Profil de l'utilisateur N°424

ID_utilisateur	Réalisateur	Genre	Préférence
424	Spielberg Steven	Action	0,81
424	Spielberg Steven	Adventure	0,87
424	Spielberg Steven	Comedy	0,75
424	Spielberg Steven	Drama	0,87
424	Spielberg Steven	Family	0,75
424	Spielberg Steven	Fantasy	0,75
424	Spielberg Steven	Horror	0,87
424	Spielberg Steven	Sci-Fi	0,83
424	Spielberg Steven	Thriller	0,85
424	Zemeckis Robert	Action	0,66
424	Zemeckis Robert	Adventure	0,66
424	Zemeckis Robert	Comedy	0,68
424	Zemeckis Robert	Drama	0,75
424	Zemeckis Robert	Romance	0,75
424	Zemeckis Robert	Sci-Fi	0,62
424	Zemeckis Robert	Western	0,25

Nous constatons d'après le tableau 8 que l'utilisateur N°424 n'a vu que des films des deux réalisateurs *Spielberg Steven* et *Zemeckis Robert*.
Imaginons maintenant que deux nouveaux films viennent de sortir.
Le film 1 est décrit comme suit : Film(ID_Film =480, Titre= Jurassic Park, Genres={Action, Aventure, Horreur, Sci-Fi, Thrilleur}, Réalisateur= Spielberg Steven).
Le film 2 est décrit comme suit : Film(ID_Film =1610, Titre= Hunt for Red, Genres ={Action ,Aventure, Thrilleur}, Réalisateur= McTiernan John).

Nous remarquons que le réalisateur du film Jurassic Park « Spielberg Steven» figure déjà dans le profil de l'utilisateur. Cela nous permet de calculer la similarité globale entre le profil utilisateur et le descripteur du film en tenant compte des préférences des genres relatifs au réalisateur « Spielberg Steven», $sim_{globale}$(ID_utilisateur = 424, ID_Film = 480) = 0,72. Enfin nous comparons cette prédiction à un seuil prédéfini pour recommander ou non le film *Jurassic Park* à l'utilisateur N°424.

Afin d'estimer l'intérêt du film *Hunt for Red* par rapport à l'utilisateur N°424, le contexte Réalisateur « McTiernan John » doit être résolu. Ceci revient à trouver qui parmi *Spielberg Steven* et *Zemeckis Robert* est le plus similaire à *McTiernan John*, ce à quoi la matrice de similarités permet de répondre. Il est à noter que cette matrice contient 427 réalisateurs ayant réalisé un total de 4141 films. Le tableau 9 montre un fragment de cette matrice.

Tableau 9. Fragment de la matrice de similarités entre réalisateurs

	Spielberg Steven	**McTiernan John**	***Zemeckis Robert***	**…**
Spielberg Steven	1	0,66	0,56	…
McTiernan John	0,66	1	0,50	…
Zemeckis Robert	0,56	0,50	1	…
…	…	…	…	…

A partir du Tableau 9, nous déduisons facilement que le réalisateur *McTiernan John* et plus similaire au réalisateur *Spielberg Steven* que ne l'est *Zemeckis Robert*. En effet, Sim(Spielberg Steven, McTiernan John) = 0,66 > Sim(Zemeckis Robert, McTiernan John) = 0,50. Par conséquent, le profil de l'utilisateur N°424 sera enrichi par des préférences contextuelles prédites et corrigées à partir de celles relatives au réalisateur *Spielberg Steven* en multipliant ces préférences par la valeur de la similarité sim(Spielberg Steven, McTiernan John) = 0,66 (voir tableau 10).

Tableau 10. Enrichissement du profil utilisateur

ID_utilisateur	**Réalisateur**	**Genre**	**Préférence**
424	McTiernan John	Action	0,81*0,66
424	McTiernan John	Adventure	0,87*0,66
424	McTiernan John	Comedy	0,75*0,66
424	McTiernan John	Drama	0,87*0,66
424	McTiernan John	Family	0,75*0,66
424	McTiernan John	Fantasy	0,75*0,66
424	McTiernan John	Horror	0,87*0,66
424	McTiernan John	Sci-Fi	0,83*0,66
424	McTiernan John	Thriller	0,85*0,66
424	Spielberg Steven	Action	0,81
424	Spielberg Steven	Adventure	0,87
424	Spielberg Steven	Comedy	0,75

424	Spielberg Steven	Drama	0,87
424	Spielberg Steven	Family	0,75
424	Spielberg Steven	Fantasy	0,75
424	Spielberg Steven	Horror	0,87
424	Spielberg Steven	Sci-Fi	0,83
424	Spielberg Steven	Thriller	0,85
424	Zemeckis Robert	Action	0,66
424	Zemeckis Robert	Adventure	0,66
424	Zemeckis Robert	Comedy	0,68
424	Zemeckis Robert	Drama	0,75
424	Zemeckis Robert	Romance	0,75
424	Zemeckis Robert	Sci-Fi	0,62
424	Zemeckis Robert	Western	0,25

Finalement, la similarité du film *Hunt for Red* par rapport à l'utilisateur N°424 est calculée comme suit : $sim_{globale}$(ID_utilisateur=424, ID_Film=1610) = 0,82
En comparant ce score d'intérêt à un score prédéfini, le système peut décider si le film *Hunt for Red* est à recommander ou non à l'utilisateur N°424.

Nous présentons dans la section suivante notre ontologie ainsi que la méthode suivie pour son élaboration.

7. Conception de l'ontologie de l'application

Dans cette thèse, nous proposons un système de recommandation ontologique basé sur le filtrage à base de contenu. Les concepts servant à modéliser les profils utilisateurs et les descripteurs de contenus de notre système sont organisés dans une ontologie que nous avons créée dans [37]. Il est à noter qu'un certain nombre de principes avaient été dégagés pour l'obtention d'une ontologie cohérente [22] et que cette dernière a été créée pour le seul cas de recommandation des films.

7.1 Méthodologie de construction

Pour construire l'ontologie, nous avons adopté les trois étapes ci-dessous :

- Spécifier les termes à collecter et les tâches à effectuer en utilisant cette ontologie. L'ontologie est construite dans l'esprit de fournir les recommandations les plus pertinentes à l'utilisateur.

- Organiser les termes en utilisant les métas catégories : concepts, relations, attributs, etc.

- Affiner l'ontologie et la structurer selon des principes de modularité et d'organisation hiérarchiques.

7.2 Respect des principes de construction

En ce qui concerne la construction de notre ontologie, « Ontologie de films », nous avons respecté les cinq critères cités ci-après:

La clarté : Pour répondre à ce principe, tous les termes utilisés dans cette ontologie ont été choisis d'une façon objective et claire : tous les genres de films qui peuvent exister sont présents dans notre ontologie.

La cohérence : Rien qui ne puisse être inféré de l'ontologie ne doit entrer en contradiction avec les définitions des concepts. Par exemple, à partir du concept « Aventure » on ne peut pas inférer des films de documentaire.

L'extensibilité : Il est possible d'ajouter de nouveaux concepts ou individus sans avoir à toucher aux fondations de l'ontologie et de l'application. Par exemple, le fait d'ajouter le film « TITANIC » au concept romance ne change en rien la structure de notre ontologie.

Un Biais d'encodage minimal : Notre ontologie est conceptualisée indépendamment de tout langage de programmation. En effet cette ontologie est exploitée avec le langage JAVA, mais on aurait pu choisir un autre langage.

Un Engagement ontologique minimal : l'ontologie ne comporte pas des connaissances supplémentaires sur le monde à modéliser.

7.3 Représentation hiérarchique de l'ontologie conceptuelle

L'ontologie de ce projet a été conçue avec Protégé. Elle est structurée en une hiérarchie de genres de films dont la racine est présentée par le concept « Film » (voir Figure 31).

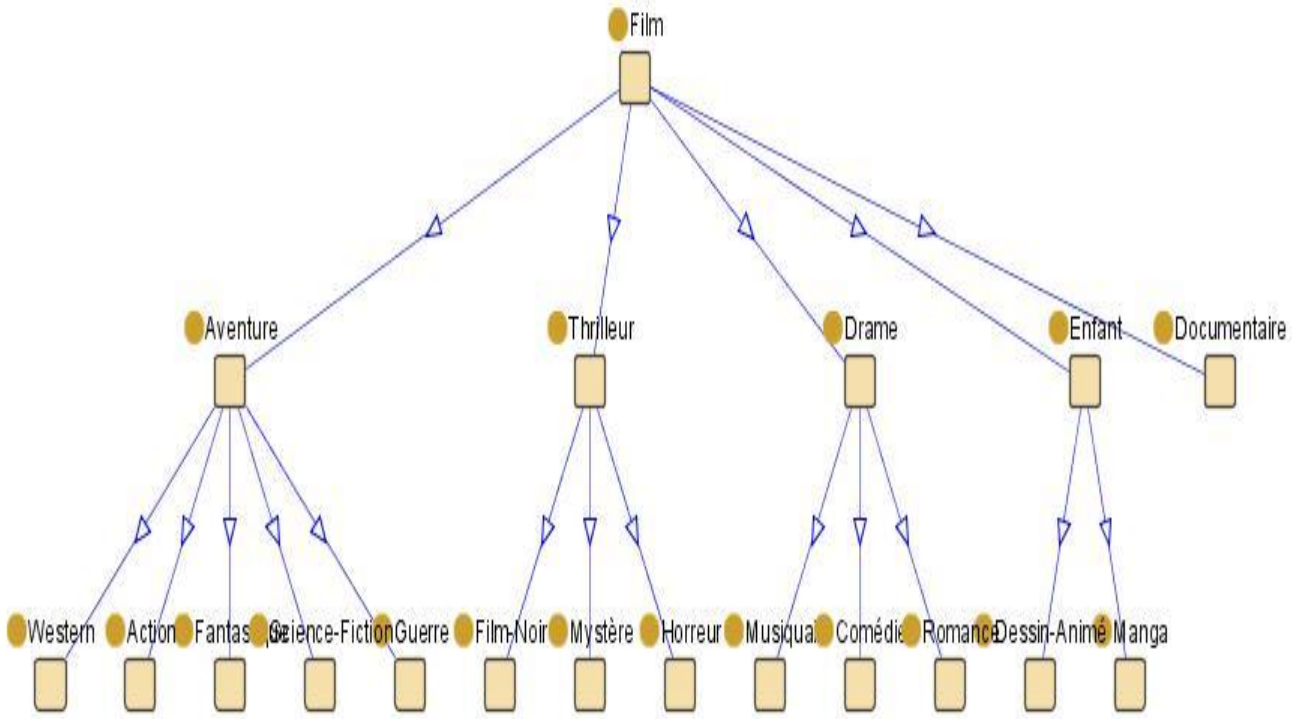

Figure 31. Représentation hiérarchique de l'Ontologie

8. Environnement de développement

Nous avons développé notre application sur une machine AMD Atlon 64 bits avec une vitesse de 1.60Ghz, dotée d'une capacité mémoire de 3GB DDR2, sous Windows 7. L'application a été réalisée avec le langage de programmation Java. Pour la base de données nous avons utilisé Oracle 10g Express Edition.

8.1 Oracle

Oracle [31] est un SGBD (Système de Gestion de Base de Donnés) édité par la société du même nom (Oracle Corporation), leader mondial de bases de données. Sa force principale est la portabilité de sa base de données. Aujourd'hui, il y a un groupe de quelques machines auxquelles Oracle accorde une priorité. Windows en fait partie, au même titre que Linux, HP/UX (Hewlett Packard), Solaris (Sun) et Aix (IBM). C'est-à-dire que toute sortie d'une nouvelle version d'Oracle sera synchronisée de manière prioritaire pour l'ensemble de ces plates-formes.

Les fonctionnalités d'oracle

Oracle est un SGBD permettant d'assurer :

- La définition et la manipulation des données
- La cohérence des données
- La confidentialité des données
- L'intégrité des données
- La sauvegarde et la restauration des données
- La gestion des accès concurrents

Les composants d'oracle

Outre la base de données, la solution oracle est un véritable environnement de travail constitue de nombreux logiciels.
On peut classer les outils d'oracle selon diverses catégories :

- Les outils d'administration
- Les outils de développement
- Les outils de communication
- Les outils de génie logiciel
- Les outils d'aide à la décision

Communication Java-Oracle(JDBC)

JDBC (Java DataBase Connectivity) est une interface de programmation créée par Sun Microsystems -depuis racheté par Oracle Corporation-, pour les programmes utilisant la plateforme Java. Elle permet aux applications Java d'accéder par le biais d'une interface commune à des sources de données pour lesquelles il existe des pilotes JDBC.

8.2 Langage de programmation java

Le langage Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.
La particularité principale de Java est que les logiciels écrits dans ce langage sont très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS

ou GNU/Linux, avec peu ou pas de modifications. C'est la plate-forme qui garantit la portabilité des applications développées en Java.

Le langage reprend en grande partie la syntaxe du langage C++, très utilisé par les informaticiens. Néanmoins, Java a été épuré des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que les pointeurs et références, et l'héritage multiple remplacé par l'implémentation des interfaces. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.)

Java permet de développer des applications client-serveur. Côté client, les applets sont à l'origine de la notoriété du langage. C'est surtout côté serveur que Java s'est imposé dans le milieu de l'entreprise grâce aux servlets, le pendant serveur des applets, et plus récemment les JSP (JavaServer Pages) qui peuvent se substituer à PHP, ASP et ASP.NET.

Java a donné naissance à un système d'exploitation (JavaOS), à des environnements de développement (eclipse/JDK), des machines virtuelles (MSJVM, JRE) applicatives multi plates-formes (JVM), une déclinaison pour les périphériques mobiles/embarqués (J2ME), une bibliothèque de conception d'interface graphique (AWT/Swing), des applications lourdes (Jude, Oracle SQL Worksheet, etc.), des technologies web (servlets, applets) et une déclinaison pour l'entreprise (J2EE). La portabilité du code Java est assurée par la machine virtuelle. JRE – la machine virtuelle qui effectue la traduction et l'exécution du bytecode en code natif – supporte plusieurs processus de compilation (à la volée/bytecode, natif). La portabilité est dépendante de la qualité de portage des JVM sur chaque OS.

Aujourd'hui, Java (sous la forme de JavaFx) trouve une nouvelle niche dans la création d'applications RIA (Rich Internet Applications), des applications qui proposent des fonctionnalités, notamment des interfaces, plus évoluées à la fois sur Internet et sur les téléphones portables. Le langage JavaFx est un langage agile dérivé de Java, sous le contrôle d'Oracle, qui met à profit la portabilité de Java ainsi que les vastes bibliothèques déjà disponibles dans le langage Java.

8.2.1 NetBeans

NetBeans [29] est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).
Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java.
NetBeans constitue par ailleurs une plate forme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plate forme (voir figure 32).

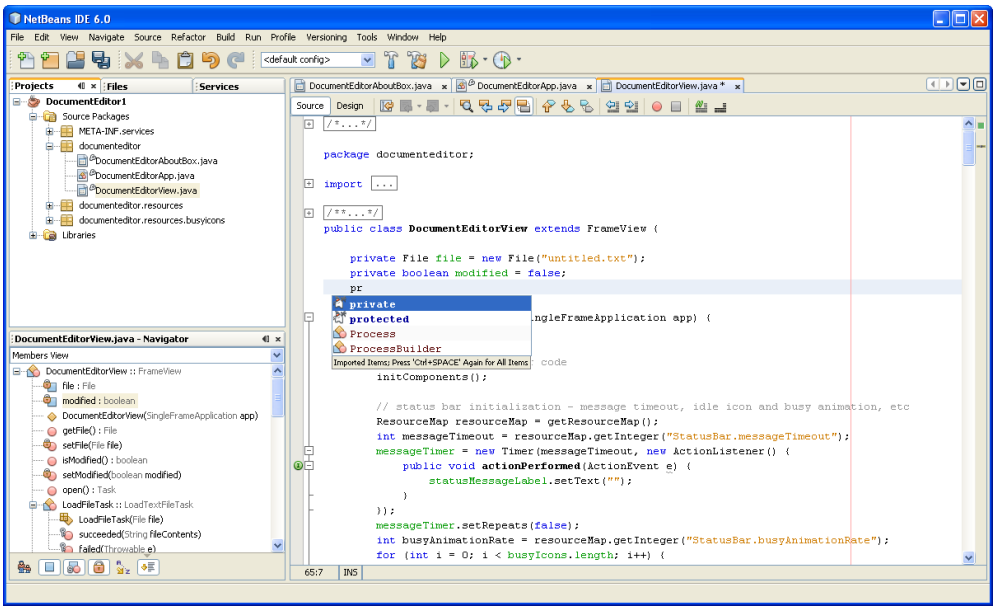

Figure 32. Capture écran de l'éditeur NetBeans

8.3 Protégé

Est un éditeur d'ontologie open source, disponible à l'adresse : http://protege.standford.edu (voir figure 33). Il a été développé au département d'Informatique Médicale de l'Université de Stanford.

L'éditeur d'ontologie Protégé « version 3.4 beta », a été utilisé pour éditer l'ontologie de ce projet dans l'objectif de générer automatiquement le code OWL (Ontology Web Language) correspondant, Il est à noter que « Protégé » offre beaucoup de fonctionnalités, qui n'ont pas été toutes utilisées dans le présent travail.

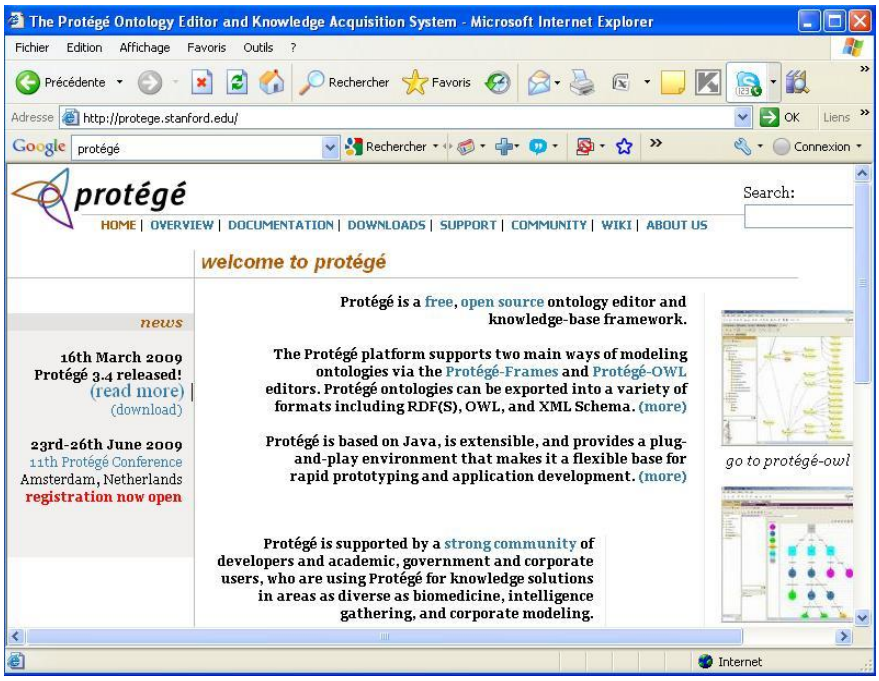
The Protégé Ontology Editor and Knowledge Acquisition System - Microsoft Internet Explorer
Fichier Edition Affichage Favoris Outils ?
Précédente Rechercher Favoris
Adresse http://protege.stanford.edu/
Google protégé Rechercher Connexion
protégé
Search:
HOME | OVERVIEW | DOCUMENTATION | DOWNLOADS | SUPPORT | COMMUNITY | WIKI | ABOUT US
welcome to protégé
news
16th March 2009
Protégé 3.4 released!
(read more)
(download)
23rd-26th June 2009
11th Protégé Conference
Amsterdam, Netherlands
registration now open
Protégé is a free, open source ontology editor and knowledge-base framework.
The Protégé platform supports two main ways of modeling ontologies via the Protégé-Frames and Protégé-OWL editors. Protégé ontologies can be exported into a variety of formats including RDF(S), OWL, and XML Schema. (more)
Protégé is based on Java, is extensible, and provides a plug-and-play environment that makes it a flexible base for rapid prototyping and application development. (more)
Protégé is supported by a strong community of developers and academic, government and corporate users, who are using Protégé for knowledge solutions in areas as diverse as biomedicine, intelligence gathering, and corporate modeling.
go to protégé-owl
Internet

Figure 33. Le site officiel du logiciel Protégé

8.4 Jena

Jena est un Framework Java pour construire des applications de Web sémantique. Il fourni un ensemble d'outils (une API) open source développé par *HP Labs semantic Web programme.*

Jena est disponible à l'adresse **:** http://jena.sourceforge.net/ La figure 34 montre la page d'accueil du site Jena.
Pour notre projet, nous avons utilisé la version 2.7 qui est la version la plus récente de Jena.

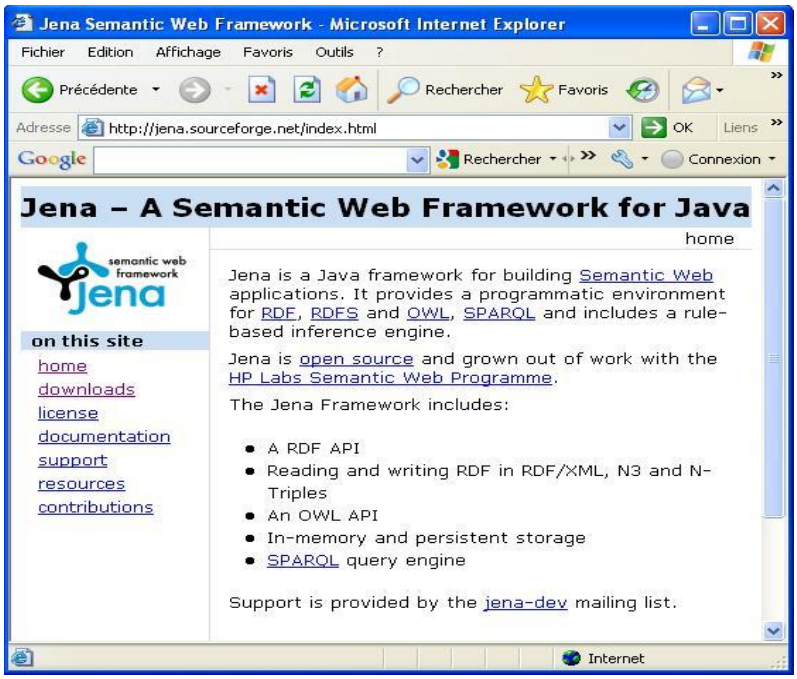

Figure 34. La page d'accueil du site

Après la génération du code OWL correspondant à notre ontologie, nous avons eu besoin de l'exploiter par notre logiciel codé en JAVA. La connexion entre un code OWL et un programme JAVA nécessite l'utilisation d'une bibliothèque (ou API) appelée Jena. Cette bibliothèque ne peut fonctionner correctement qu'avec la version 1.7 de JDK (Java Développement Kit).
Pour ajouter la librairie Jena dans NetBeans, il faut aller au menu Outils/configurer librairies et cliqué sur le bouton « Ajouter » (voir figure 35).

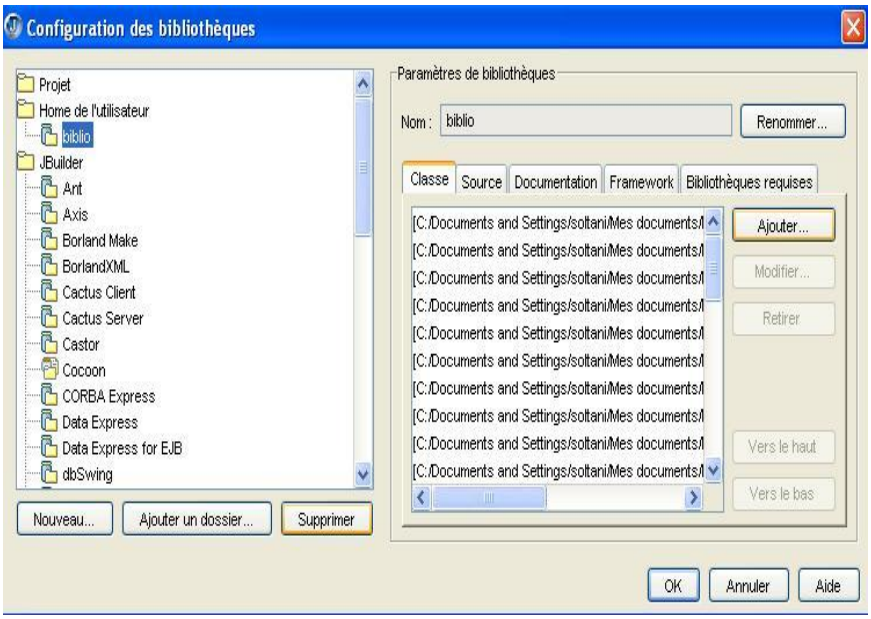

Figure 35. Ajout de la librairie Jena

9. Conclusion

Dans ce chapitre, nous avons exposé et expliqué les différentes phases suivies pour la conception et la réalisation de notre système de recommandation sensible aux contextes.
En effet, nous avons explicité le modèle de données utilisé, la méthode de création des profils utilisateur ainsi que la méthode de résolution du contexte. Nous avons également, présenté les deux algorithmes que nous avons élaborés. Le premier nous a permis la création automatique des profils en exploitant les logs d'interaction; le second a permis l'extraction du contexte intrinsèque directement à partir des contenus identifiés comme candidats potentiels à la recommandation. Ensuite, nous avons présenté une instanciation de nos algorithmes en les appliquant dans le domaine cinématographique en choisissant le "réalisateur" comme contexte intrinsèque. Enfin, nous avons décrit l'environnement de travail et les outils mis en jeu pour la réalisation de ce système.

Dans le chapitre suivant, nous allons présenter les différentes phases adoptées pour l'évaluation de notre système de recommandation contextuel.

CHAPITRE IV

Expérimentations et Evaluation

1. Introduction

Après avoir détaillé la partie conception de notre système, nous consacrons ce chapitre à son évaluation (une partie de cette évaluation a été présentée dans [6]). Nous commençons d'abord par décrire la base de données utilisée pour l'évaluation, ainsi que la méthodologie suivie pour la construction des tables et le calcul de prédiction. Nous enchaînons en présentant l'évaluation en adoptant les deux métriques les plus utilisées dans le domaine de la recherche d'information, à savoir, la précision et le rappel. Enfin, nous terminons avec une interprétation des résultats de l'évaluation du système avec contexte en les comparant à ceux de l'évaluation du SR traditionnel (sans contexte).

2. MovieLens et IMDB

Pour l'évaluation nous nous sommes basé sur la plateforme de test réalisée au cours du projet APMD (Accès Personnalisé à des Masses de Données, URL: http://apmd.prism.uvsq.fr) et qui regroupe deux bases de données sur les films, à savoir, IMDB (Internet Movie Database, URL: http://www.imdb.com) qui fournit les caractéristiques des films (titre, genre, acteurs, réalisateur, etc.) et MovieLens (URL: http://www. movielens.umn.edu) qui fournit les notes données par des utilisateurs à certains films. Ces notes sont comprises entre 1 et 5.

La répartition des notes données par les utilisateurs est la suivante :

- Défavorables : 5% (pour la note 1), 10% (pour la note 2).
- 35 % des évaluations effectuées sont représentées par des notes de 4
- 26 % par des notes de 3
- 23 % par des notes de 5

La figure 36 montre la répartition des notes dans MovieLens

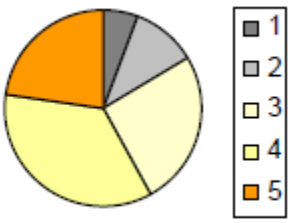

Figure 36. Les répartitions des évaluations

Nous nous sommes basés également, sur la représentation et la définition des bases de données présentées pour l'extraction des données de MovieLens et IMDB [33].

Nous présentons dans ce qui suit quelques tables utilisées qui nous ont aidés dans la réalisation et l'évaluation au cours de ce projet.

La table I_movies contient les titres des films tirés des deux bases MovieLens et IMDB. Cette table contient les 3340 films avec leurs titres. La table I_moviegenres contient chacun des films avec les genres correspondants. Nous avons retenu 25 genres de films pour l'expérimentation. La table I_userratings contient tous les utilisateurs et tous les films avec leurs notes (rating) extraites de la base de données MovieLens. Cette table compte environ 6000 utilisateurs. La table Réalisateurs contient chacun des films avec le (ou les) réalisateur (s) correspondant (s). La constitution de ces tables est donnée dans les quatre tables suivantes.

I_movies

Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
MOVIEID	NUMBER(5,0)	Yes	-	-
TITLEMOVIELENS	VARCHAR2(100)	Yes	-	-
TITLEIMDB	VARCHAR2(250)	Yes	-	-
				1 - 3

I_moviegenres

Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
MOVIEID	NUMBER(5,0)	Yes	-	-
GENRE	VARCHAR2(15)	Yes	-	-
				1 - 2

I_userratings

Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
USERID	NUMBER(5,0)	Yes	-	-
MOVIEID	NUMBER(5,0)	Yes	-	-
RATING	NUMBER(1,0)	Yes	-	-
TIMESTAMP	NUMBER(10,0)	Yes	-	-
C1	NUMBER(1,0)	Yes	-	-
C2	NUMBER(1,0)	Yes	-	-
C3	NUMBER(1,0)	Yes	-	-
C4	NUMBER(1,0)	Yes	-	-
C5	NUMBER(1,0)	Yes	-	-
MOD	NUMBER(1,0)	Yes	-	-
				1 - 10

I_director

Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
MOVIEID	NUMBER(5,0)	Yes	-	-
DIRECTOR	VARCHAR2(15)	Yes	-	-
				1 - 2

De la table I_userratings, nous avons extrait la table T-usernote représentée par la figure 37. Cette dernière contient 21 utilisateurs ayant regardé les mêmes 100 films ainsi que leurs notes.

Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
USERID	NUMBER(5,0)	Yes	-	-
MOVIEID	NUMBER(5,0)	Yes	-	-
NOTE	NUMBER(2,0)	Yes	-	-
				1 - 3

Figure 37. Table T-usernote

La figure 38 montre la table Training_set, cette dernière contient une partie des films choisis aléatoirement à partir de la table T_usernote. Elle détaille pour les 21 utilisateurs, les genres de chaque film ainsi que la propagation de la note sur ses genres.

Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
USERID	NUMBER(5,0)	Yes	-	-
MOVIEID	NUMBER(5,0)	Yes	-	-
NOTE	NUMBER(2,0)	Yes	-	-
				1 - 3

Figure 38. Table Training _set

La figure 39 montre la table du profil utilisateur, cette dernière contient les 21 utilisateurs, les genres des films, et la préférence de l'utilisateur pour chaque genre. Les préférences sont déduites par calcul.

Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
USERID	NUMBER(5,0)	Yes	-	-
GENRE	VARCHAR2(15)	Yes	-	-
PREFERENCE	FLOAT	Yes	-	-
				1 - 3

Figure 39. Table Profil_user

Dans la figure 40, nous avons la table Result_set, elle contient le complément de la table Training_set par rapport à la table T_usernote. Elle est composée des 21 utilisateurs, des (100-i) films (tel que i représente le nombre de films dans la table Training_set), de la note attribuée par l'utilisateur à chaque film et des prédictions qui sont calculées par l'application.

Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
USERID	NUMBER(5,0)	Yes	-	-
MOVIEID	NUMBER(5,0)	Yes	-	-
NOTE	NUMBER(2,0)	Yes	-	-
PREDICTION	FLOAT	Yes	-	-
				1 - 4

Figure 40. Table Result_set

La signification des différents champs des tables utilisées est donnée dans le (tableau 11).

Tableau 11. Dictionnaire de Base de Données

Champs	Signification
USERID	L'identificateur de l'utilisateur
MOVIEID	L'identificateur de film
NOTE	La note donnée par l'utilisateur à un film
PREDICTION	Les prédictions calculées par l'application
GENRE	Les genres de film
DIRECTOR	Le réalisateur de film

Remarque :

Les tables représentées ci-dessus sont utilisées pour l'évaluation du système de recommandation sans contexte. Pour l'évaluation du système de recommandation avec contexte, nous devons effectuer des jointures entre quelques unes de ces tables. Nous verrons cela dans la section suivante.

3. Evaluation du système de recommandation

L'évaluation des algorithmes de recommandation est une tâche complexe. En effet, les systèmes ont souvent été évalués selon des critères différents. Parmi ces métriques, nous pouvons citer la précision et le rappel. Nous présentons dans ce qui suit les différentes étapes par lesquelles nous sommes passés pour évaluer notre système.

3.1 Construction des tables

Comme nous l'avons déjà mentionné ci-dessus, de notre benchmark sont extraits une partie **Training_set** pour l'apprentissage des profils utilisateurs et une autre partie **Result_set** qui sert à tester le processus de recommandation.

Ce benchmark est constitué de 21 utilisateurs qui ont regardés les même 100 films. Il a été utilisé pour remplir la table **T_usernote** (présentée à la figure 37). Une partie de cette dernière est montrée dans le tableau 12.

Tableau 12. Un extrait de la table T_usernote

USERID	MOVIEID	NOTE
424	589	4
424	780	3
424	923	5
:	:	:
:	:	:
1181	1247	2
1181	1304	4
:	:	:
:	:	:
5954	923	4

5954	1193	5
5954	1208	5
:	:	:
:	:	:

A partir de la table **T_usernote,** nous avons rempli les tables **Training_set** et **Result_set** de façon aléatoire en utilisant une procédure aléatoire. Cette procédure permet de donner les répartitions suivantes : {(70,30) ;(60,40) ;(50,50)…etc.}. Par exemple, pour le premier couple (70,30), le premier composant constitue les 70 films qui servent à remplir la table **Training_set** et le second, c'est-à-dire les 30 restants parmi les 100 films initiaux, constitue l'ensemble des films qui servent à remplir la table **Result_set**. La table **Training_set,** pour cet exemple contiendra 21 utilisateurs (représentés par leurs identificateurs USERID), 70 films pour chaque utilisateur (représentés par leurs identificateurs MOVIEID), les genres associés à chacun des films, le réalisateur (ajouté à partir de la table Réalisateurs) et la note du film propagée sur les genres du film.
Le tableau 13 montre une partie de la table **Training_set.** Par exemple, nous pouvons constater sur ce tableau, que l'utilisateur portant l'identificateur 424 a regardé les films portant les identificateurs 110 et 296. Le film 110 ayant comme réalisateur Gibson Mel et comme genres Action, Biography, War, History, Drama, a eu la note 4 par l'utilisateur 424. Cette note a été propagée sur ses genres.

Tableau 13. Un extrait de la table Training_set

USERID	MOVIEID	GENRE	DIRECTOR	NOTE
424	110	Action	Gibson Mel	4
424	110	Biography	Gibson Mel	4
424	110	War	Gibson Mel	4
424	110	History	Gibson Mel	4
424	110	Drama	Gibson Mel	4
424	296	Crime	Tarantino Quentin	5
424	296	Drama	Tarantino Quentin	5
:	:	:		:
:	:	:		:
5954	296	Crime	Tarantino Quentin	5
5954	296	Drama	Tarantino Quentin	5

La table **Result_set,** pour la répartition (70, 30) contiendra 21 utilisateurs (représentés par leurs identificateurs USERID), 30 films pour chaque utilisateur (représentés par leurs identificateurs MOVIEID), les notes associées à chacun des films et la colonne prédiction qui sert pour le calcul de la similarité entre les genres d'un film et les genres préférés par un utilisateur. Cette colonne est initialement vide.
Le tableau 14 montre une partie de la table **Result_set.** Par exemple, nous pouvons constater sur ce dernier, que l'utilisateur portant l'identificateur 424 a regardé les films portant les identificateurs 349, 356, 380, 457 auxquels il a attribué les notes 3, 4, 4, 4 respectivement.
La table **Result_set** sert pour le calcul des prédictions afin de tester le processus de recommandation des films aux utilisateurs.

Tableau 14. Un extrait de la table Result_set

USERID	MOVIEID	NOTE	PREDICTION
424	349	3	-
424	356	4	-
424	380	4	-
424	457	4	-
:	:	:	:
:	:	:	:
5954	349	2	-
5954	356	4	-
5954	380	1	-
5954	457	3	-
:	:	:	:
:	:	:	:

L'étape suivante consiste à construire la table **Classic Profil_user.** Cette dernière permet de générer le profil d'un utilisateur à partir de la table **Training_set.** Elle contient les 21 utilisateurs, les genres des films tirés de la table **Training_set** correspondant à une répartition aléatoire donnée, et une colonne PREFERENCE correspondant à la préférence d'un utilisateur pour chaque genre.
La valeur de la préférence d'un genre pour un utilisateur donné est obtenue par le calcul de la moyenne des notes de ce genre dans la table **Training_set.** Le tableau 15 présente une partie de la table **Classic_Profil_User** qui sera utilisé pour l'évaluation sans contexte**.** Elle montre plus exactement les préférences de l'utilisateur 424.

Tableau 15. Un extrait de la table Classic_Profil_user

USERID	GENRE	PREFERENCE
424	Mystery	4,14
424	Sci-Fi	2,81
424	Animation	4
424	Biography	4,66
424	Comedy	4
424	Family	4,2
424	Musical	4
424	Thriller	1,2
424	Romance	3
424	Sport	4,2
424	Western	3,21
424	Adventure	4,04
424	Drama	4,3
424	Horror	4,66
424	Action	4,10
424	War	2,16
424	Fantasy	4,27
424	Crime	4,55
424	History	4

Le tableau ci-dessous (tableau 16) montre une partie de la table **Contextual_Profil_User** et plus exactement l'utilisateur 424 qui est construit de la même manière que le profil précédent, sauf que la préférence est calculée par rapport aux genres et aux réalisateurs d'un film pour un utilisateur donné.

Tableau 16. Un extrait de la table Contextual_Profil_User

USERID	DIRECTOR	GENRE	PREFERENCE
424	Altman Robert	Comedy	4.0
424	Altman Robert	Drama	4.0
424	Altman Robert	War	4.0
424	Besson Luc	Action	3.0
424	Besson Luc	Adventure	3.0
424	Besson Luc	Sci-Fi	3.0
424	Besson Luc	Thriller	3.0
424	Burton Tim	Action	5.0
424	Burton Tim	Crime	5.0
424	Burton Tim	Fantasy	4.5
424	Burton Tim	Thriller	5.0
424	Burton Tim	Comedy	4.0
424	Burton Tim	Horror	4.0
424	Cameron James	Action	4.6666665
424	Cameron James	Adventure	4.0
424	Cameron James	Drama	4.0
424	Cameron James	Sci-Fi	4.6666665
424	Cameron James	Thriller	4.6666665

Dans ce qui suit nous montrons comment calculer la préférence de l'utilisateur 424 pour le genre Action et le réalisateur Cameron James.

Tableau 17. Un extrait de la table Training_set pour le genre Action

USERID	MOVIEID	GENRE	DIRECTOR	NOTE
424	1127	Action	Cameron James	4
:	:	:		:
:	:	:		:
424	1200	Action	Cameron James	5
:	:	:		:
:	:	:		:
424	1240	Action	Cameron James	5
:	:	:		:
:	:	:		:

La préférence de l'utilisateur 424 pour le genre Action = (4+5+5)/3= 4,66. Cette dernière représente la valeur de la préférence du genre Action pour l'utilisateur 424 en prenant en compte le contexte intrinsèque défini par le réalisateur Cameron James (voir tableau 17).

Un changement d'échelle s'avère nécessaire à ce stade du calcul, car les scores (préférences) doivent être compris entre '0' et '1'. Nous utilisons, pour cela, la formule : X-X0 / X1-X0, avec X0 = 1 et X1 = 5. Par exemple la préférence pour l'utilisateur 424 pour le genre Musical, qui est égale à 4,2 sera transformée en : 4,2-1 / 5-1 = 3,2/4 = 0,8.

Afin de calculer la similarité entre les réalisateurs, que nous allons expliquer son utilité plus loin, nous avons besoin des profils réalisateurs. Nous représentons chaque réalisateur (contexte intrinsèque) par un vecteur de genres pondérés, Réalisateur (<Action,0.7>, ... ,<Drame,0.2>). La pondération (score compris entre 0 et 1) associée à chaque genre renseigne sur le pourcentage de films de ce genre réalisés par le réalisateur en question (nombre de film d'un certain genre/nombre total de films réalisés). Par exemple <Action, 0.7> renseigne sur le fait que 70% des films réalisés par le réalisateur sont des films d'Action. Le tableau 18 montre une partie des vecteurs représentant les réalisateurs avec les attributs (réalisateur, genre, et moyenne).

Tableau 18. Un extrait de la table profil réalisateur

DIRECTOR	GENRE	AVERAGE
Eastwood Clint	Action	0.25
Eastwood Clint	Adventure	0.16
Eastwood Clint	Comedy	0.25
Eastwood Clint	Crime	0.33
..	..	..
..	..	..
Zemeckis Robert	Action	0.44
Zemeckis Robert	Animation	0.11
Zemeckis Robert	Comedy	0.77
..	..	..
..	..	..

Après avoir calculé les vecteurs de tous les réalisateurs, nous pouvons procéder au remplissage de la matrice de similarité réalisateurs / réalisateurs. Cette dernière est une matrice carrée, symétrique contenant en ligne et en colonne tous les réalisateurs de la base de données (voir tableau 19). Nous utilisons la corrélation de Pearson pour calculer la similarité entre ses réalisateurs.

Tableau 19. Un extrait de la matrice réalisateur/réalisateur

	Abraham Adam	Bay Michael	Landis John	..
Abraham Adam	1	0.24	0.9	..
Bay Michael	0.24	1	0.52	..
Landis John	0.9	0.52	1	..
..	..	..	..	..

Dans le tableau 19, nous remarquons que la similarité Sim (Abraham Adam, Abraham Adam) = 1 (car le réalisateur est similaire à lui-même). Sim (Bay Michael, Landis John) = 0.52 indique que le réalisateur *Bay Michael* est similaire au réalisateur *Landis John* avec une valeur de 0.52

3.2 Calcul de la prédiction

Pour pouvoir faire l'évaluation du prototype, il faut comparer entre les notes réelles données par l'utilisateur sur le film et la prédiction calculée par le prototype. Pour le calcul de la similarité numérique, nous utilisons la mesure de Pearson, et pour le calcule de la similarité sémantique nous utilisons la mesure Jiang & Conrath. Enfin nous combinons les deux mesure afin d'avoir de meilleurs résultats.

3.2.1 Similarité numérique

Pour calculer la prédiction dans cette mesure, nous avons besoin de deux vecteurs, le premier correspond aux genres de films à prédire (prédiction contextuelle et non contextuelle), le deuxième vecteur contient les préférences extraites de la table **Classic_ Profil_User** pour la prédiction non contextuelle. Tandis que les préférences extraites de la table **Contextual_Profil_User** sont utilisées pour la prédiction contextuelle.

Le tableau 20 est une illustration du premier vecteur pour l'utilisateur d'identificateur 424.
Le deuxième vecteur correspond à la pondération des genres associés à un film ayant l'identificateur 1387 (voir tableau 21).

Tableau 20. Illustrations du vecteur User pour le calcul de la similarité numérique

Genre / User	Thriller	Drama	Horror	Animation	Crime	Adventure	Sci-Fi	…… ……
424	0,05	0,82	0,91	0 ,75	0,88	0,76	0,45	…… ……

Tableau 21. Illustration du vecteur Film pour le calcul de la similarité numérique

Genre / Film	Thriller	Drama	Horror	Animation	Crime	Adventure	Sci-Fi	…… ……
1387	1	0	1	0	0	1	0	…… ……

3.2.2 Similarité sémantique

Le tableau 22 montre comment nous calculons la similarité sémantique entre un film dont les genres sont Drame, Musical et Comédie et un utilisateur qui préfère les genres Drame et Romance. La mesure sémantique utilisée dans cet exemple est la mesure de Jiang et Conrath.

Tableau 22. Illustrations d'une matrice pour le calcul de la similarité sémantique

Film1 / User 1	Drama	Musical	Comedy
Drama	1	0,62	0,62
Romance	0,62	1,20	1,20

Le poids de chaque matrice est calculé par la formule suivante

$$Similarity\ (V1, V2) = \frac{1}{M * N} \sum_{j=1}^{m} \sum_{i=1}^{n} simij$$

3.2.3 Similarité globale

Dans ce type de similarité, nous avons effectué une combinaison linéaire entre la mesure sémantique de Jiang et Conrath et la mesure numérique de Pearson. En utilisant la formule suivante :

Sim-glob = Coef 1 * Sim-sém de Jiang + Coef 2 * Sim-num de Pearson

Avec Coef 1 et Coef 2 = 1.

3.3 Recommandation contextuelle

Nous distinguons trois cas pour le calcul de la recommandation contexttuelle:

1. Lorsque les deux vecteurs ont le même réalisateur, nous appliquons la méthode de similarité globale.
Sim-glob = Coef 1 * Sim-sém + Coef 2 * Sim-num. Avec Coef 1 + Coef 2 = 1.
D'après nos expérimentations, Coef 1 = 0,75 et Coef 2 = 0,25 conduisent à de bonnes recommandations aux utilisateurs.

2. Lorsqu'un nouveau contexte (réalisateur) se présente à l'utilisateur (typiquement un réalisateur pour lequel l'utilisateur n'a vu aucun film auparavant). Dans ce cas, nous devons prédire les préférences qu'aurait eues l'utilisateur par rapport aux genres de films produit par ce réalisateur inconnu. Pour cela, nous devons trouver parmi les réalisateurs connus par l'utilisateur (ceux présents dans son profil), celui qui est le plus similaire au réalisateur inconnu. Pour trouver ce réalisateur inconnu, nous devons calculer tous les profils réalisateurs qui vont nous servir par la suite à trouver la similarité entre tous les réalisateurs de notre base de données. Une fois la similarité entre tous les réalisateurs est calculée, nous pouvons chercher parmi les réalisateurs figurant sur le profil utilisateur contextuel celui qui ressemble le plus au réalisateur inconnu, Par conséquent, le profil de cet utilisateur sera enrichi par des préférences contextuelles prédites et corrigées à partir de celles relatives au réalisateur le plus similaire.

3. Lorsque nous avons un film avec deux réalisateurs (un connu et l'autre inconnu) dans ce cas, nous calculons la similarité globale entre le profil utilisateur et le profil film qui ont le même réalisateur ensuite nous appliquons la méthode précédente pour le réalisateur inconnu. Enfin nous calculons la moyenne entre les deux similarités précédentes.

Nous répétons ce procédé pour tous les utilisateurs et tous les films qui existent dans la table **Result_set**. Ensuite nous sauvegardons tous ces résultats dans une table. Le tableau 23 montre un extrait de cette sauvegarde.

Tableau 23. Résultat expérimental selon la similarité globale

USERID	MOVIEID	NOTE	PREDICTION
424	1387	5	0.84
	:	:	:
	:	:	:

3.4 La précision et le rappel

Nous détaillons à travers cette section, le calcul de précision pour chacune des méthodes (méthode du seuil, méthode des Top(k)), ainsi que le calcul du rappel en utilisant la méthode du seuil.

3.4.1 La précision

Pour calculer la précision, nous avons utilisé deux méthodes. La première consiste à considérer les TOP (K) films (les K meilleurs films). La deuxième consiste à considérer un seuil de prédiction à partir duquel les films sont considérés comme recommandables.

3.4.1.1La méthode des TOP(K)

Cette méthode consiste à faire, pour un utilisateur donné, l'intersection entre les K premiers films notés effectivement par cet utilisateur et les K premiers films prédis par le prototype, et à diviser le résultat de cette intersection par le nombre K. Nous répétons ce procédé pour chacun des utilisateurs constituant la table Result_set. La moyenne de ce calcul pour tous les utilisateurs nous délivre la précision pour la valeur de K. La variation du nombre K nous permet d'obtenir d'autres valeurs de précision qui vont nous permettre de tracer la courbe de précision. La formule utilisée est la suivante :

$$précision = \frac{TopKprédits \cap TopKréels}{K}$$

Exemple :

En se basant sur le tableau 24, nous montrons comment calculer la précision pour les TOP(5) films des deux utilisateurs 424 et 549.

Tableau 24. Exemple de la méthode des TOP(K)

USERID	MOVIEID	NOTE	PREDICTION
424	1080	3	0,8078526
424	2193	3	0,790375
424	1193	5	0,9103471
424	2918	4	0,8078526
424	1208	5	0,78291196
424	367	4	0,7987779
424	2405	3	0,7932029
424	1036	5	0,81491566
424	1527	3	0,78197324
424	3421	5	0,8078526

549	923	5	0,8167297
549	1193	5	0,9696345
549	1208	5	0,8271942
549	1213	4	0,7143431
549	1219	5	0,7462347
549	1247	4	0,74693835
549	1380	5	0,6336822
549	1527	5	0,82969207
549	1958	3	0,74693835
549	2088	5	0,7576315

Les 5 premiers films notés effectivement par l'utilisateur 424 sont {1193, 1208, 1036, 3421, 2918}.
Les 5 premiers films prédits par le prototype pour l'utilisateur 424 sont {1193, 1036, 2918, 3421, 367}.
Alors la précision donnée par TOP(5) de l'utilisateur 424 est donné par ({1193, 1208, 1036, 3421, 2918} ∩ {1193, 1036, 2918, 3421, 367})/ 5 = 3/5 =0,6.

Les 5 premiers films notés effectivement par l'utilisateur 549 sont {923, 1193, 1208, 1219, 1380}.
Les 5 premiers films prédits par le prototype pour l'utilisateur 549 sont {1193, 1527, 1208, 923, 2088}.
Alors la précision donnée par TOP(5) de l'utilisateur 549 est donné par {923, 1193, 1208, 1219, 1380} ∩ {1193, 1527, 1208, 923, 2088})/ 5 = 3/5 =0,6.

Par conséquent, la précision moyenne par la méthode des TOP(5) pour ces deux utilisateurs est donnée par : (3/5+3/5) / 2= 0,6.

3.4.1.2La méthode du seuil

Cette méthode utilise un seuil pour les notes et un autre pour les prédictions. Les étapes données ci-dessous sont suivies pour le calcul de la précision :

1. Nous prenons un certain seuil pour la note (seuil-note) et un autre seuil pour la prédiction (seuil-prédiction), comme par exemple, seuil-note =3 et seuil-prédiction = 0 ,5.
2. Nous calculons la précision en utilisant la formule suivante pour chacun des utilisateurs dans la table Result_set :

$$\textbf{Précision} = \frac{\textbf{Nombre de films tel que : note} > \textbf{seuil-not} \cap \textbf{prédiction} > \textbf{seuil-prédiction}}{\textbf{Nombre de films tel que prédiction} > \textbf{seuil-prédiction}}$$

3. On fait la moyenne des précisions des 21 utilisateurs de la table Result_set.

Ces trois étapes nous délivrent le calcul de la précision en fonction de la taille de la table Training_set correspondant à la taille de la table Result_set en question. On refait les étapes 2 et 3 pour les autres tables Result_set.

3.4.2 Le rappel

Pour le calcul de cette métrique, nous procédons de la même façon avec laquelle nous avons calculé la précision par la méthode du seuil. Le seul changement réside dans le dénominateur, la formule du rappel est donnée par :

$$\textbf{\textit{Rappel}} = \frac{\textbf{\textit{Nombre de films tel que : note} > \textit{seuil-not} \cap \textit{prédiction} > \textit{seuil-prédiction}}}{\textbf{\textit{Nombre de films tel que note} > \textit{seuil-note}}}$$

Exemple :

A partir du tableau 25, nous montrons comment calculer la précision et le rappel des deux utilisateurs 424 et 549 en considérant **seuil-note=4** et **seuil-prédiction=0,75**.

Tableau 25. Exemple de la méthode du seuil

USERID	MOVIEID	NOTE	PREDICTION
424	1080	3	0,8078526
424	2193	3	0,790375
424	1193	5	0,9103471
424	2918	4	0,8078526
424	1208	5	0,78291196
424	367	4	0,7987779
424	2405	3	0,7932029
424	1036	5	0,81491566
424	1527	3	0,78197324
424	3421	5	0,8078526
549	923	5	0,8167297
549	1193	5	0,9696345
549	1208	5	0,8271942
549	1213	4	0,7143431
549	1219	5	0,7462347
549	1247	4	0,74693835
549	1380	5	0,6336822
549	1527	5	0,82969207
549	1958	3	0,74693835
549	2088	5	0,7576315

Les films qui ont un seuil-note > 4 pour l'utilisateur 424 sont {1193, 1208, 1036, 3421}.
Les films qui ont un seuil-prédiction > 0,75 pour l'utilisateur 424 sont {1080, 2193, 1193, 2918, 1208, 367, 2405, 1527, 1036, 3421}, alors la précision donnée par la méthode du seuil de l'utilisateur 424 est donnée par ({1193, 1208, 1036, 3421} ∩ {1080, 2193, 1193, 2918, 1208, 367, 2405, 1527, 1036, 3421})/ {1080, 2193, 1193, 2918, 1208, 367, 2405, 1527, 1036, 3421} ={1193, 1208, 1036, 3421}/{1080, 2193, 1193, 2918, 1208, 367, 2405, 1527, 1036, 3421} = 4/10= 0,4.
Le dénominateur 10 représente le nombre d'éléments constituant {1080, 2193, 1193, 2918, 1208, 367, 2405, 1527, 1036, 3421}

Le rappel calculé par la méthode du seuil pour l'utilisateur 424 est donné par ({1193, 1208, 1036, 3421} ∩ {1080, 2193, 1193, 2918, 1208, 367, 2405, 1527, 1036, 3421})/ {1193, 1208, 1036, 3421} = 4/4= 1.
Le dénominateur 4 représente le nombre d'éléments constituant {1193, 1208, 1036, 3421}.

Les films qui ont un seuil-note > 4 pour l'utilisateur 549 sont {923, 1993, 1208, 1219, 1380, 1527, 2088}.
Les films qui ont un seuil-prédiction > 0,75 pour l'utilisateur 549 sont ({923, 1193, 1208, 1527, 2088}, alors la précision donnée par la méthode du seuil de l'utilisateur 549 est donné par {923, 1993, 1208, 1219, 1380, 1527, 2088} ∩ {923, 1193, 1208, 1527, 2088})/ 5 = 4/5= 0,8.
Le rappel calculé par la méthode du seuil pour l'utilisateur 549 est donné par ({923, 1993, 1208, 1219, 1380, 1527, 2088} ∩ {923, 1193, 1208, 1527, 2088})/ 7 = 4/7= 0,57.
Par conséquent, la précision moyenne par la méthode du seuil pour ces deux utilisateurs est donnée par : (4/10+4/5) / 2= 0,6.
Le rappel moyen par la méthode du seuil pour ces deux utilisateurs est donné par :
(4/4+4/7) / 2= 0,78.

3.5 Interprétation de l'évaluation

Comme nous l'avons précisé précédemment, pour faire l'évaluation du système de recommandation, nous nous sommes basés sur les deux métriques : précision et rappel. Concernant la précision, nous avons utilisé les deux méthodes évoquées ci-dessus. Après les calculs de précision et de rappel, nous avons tracé les courbes correspondantes pour la méthode contextuelle et non contextuelle. Nous présentons et discutons l'allure de ces courbes dans ce qui suit.

3.5.1 Résultats de la méthode TOP(K)

Nous avons fait plusieurs expérimentations en effectuant des changements sur les mesures de similarité (sémantique, numérique et globale) ainsi que sur la valeur de K pour une taille de la table Training_set fixée. La valeur des coefficients retenus sont : (coef1=0,75 pour la similarité sémantique et coef2= 0,25 pour la similarité numérique), Les différentes valeurs de K considérées sont : 5, 10, 15, 20 ,25. Nous avons représenté les résultats d'expérimentation par des courbes. Chaque courbe nous donne la valeur de la précision en fonction de la variation de K pour une taille de Training_set précise. Les tailles de Training_set considérées sont : 30, 50, 70. Les figures 41, 42 et 43 présentent les courbes de l'approche non contextuelle, tandis que les figures 44, 45, et 46 présentent celles de l'approche contextuelle.

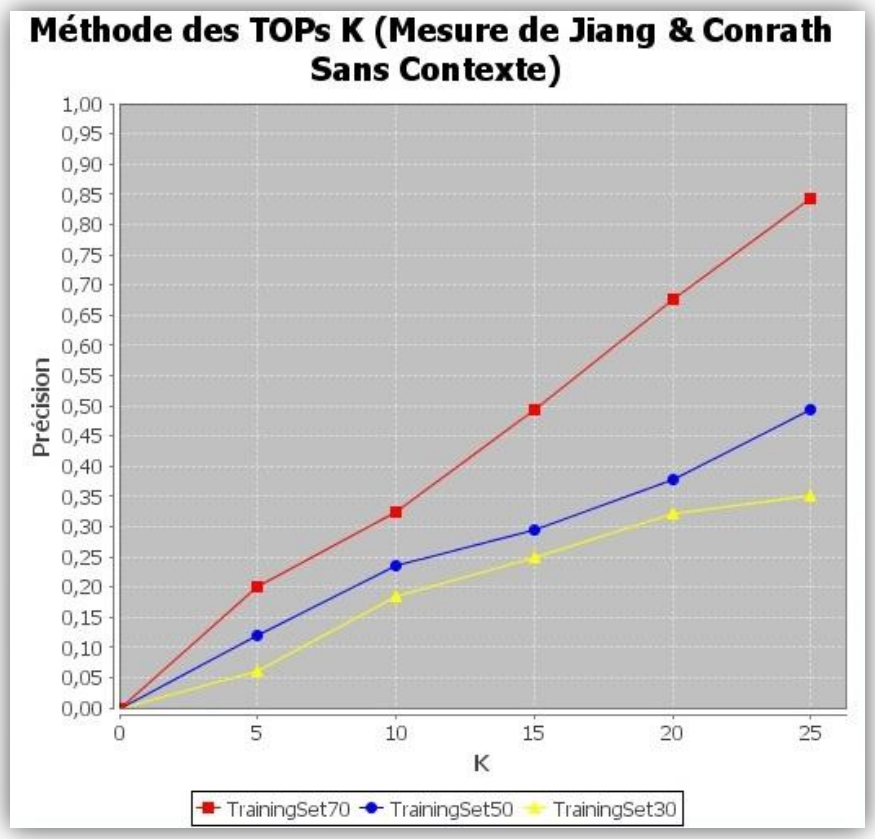

Figure 41. Méthode des TOPs k (Mesure de Jiang & Conrath Sans Contexte)

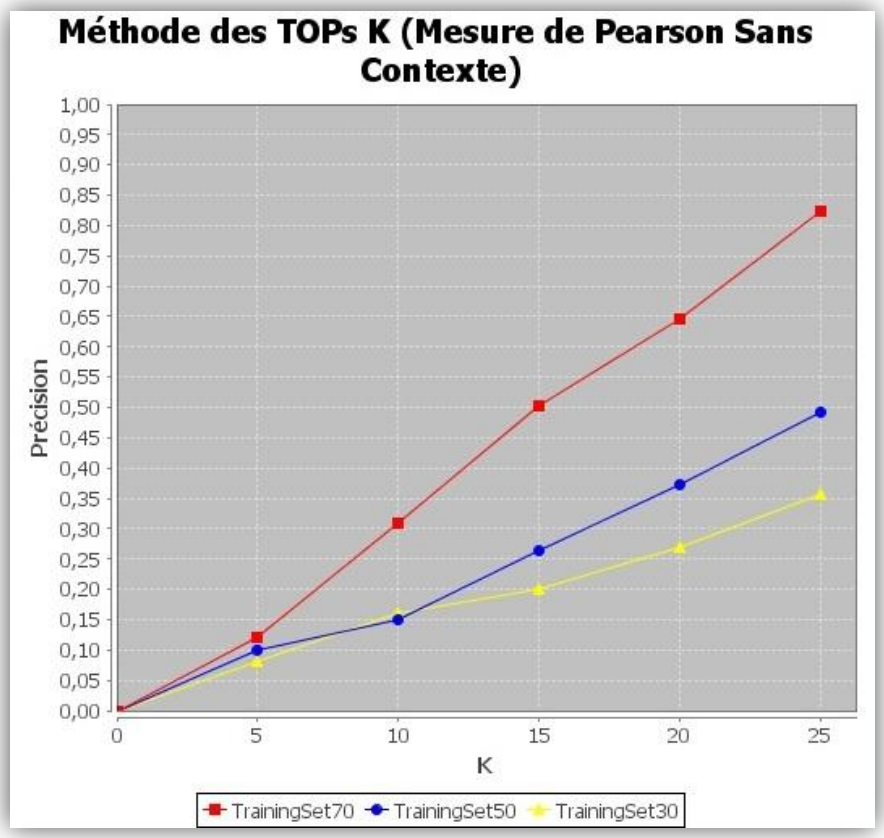

Figure 42. Méthode des TOPs k (Mesure de Pearson Sans Contexte)

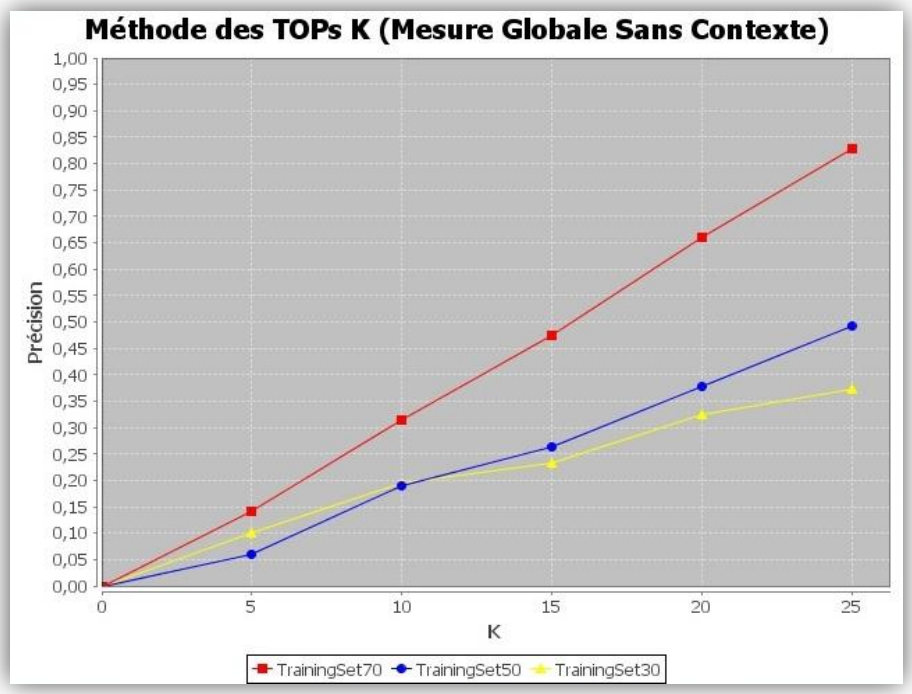

Figure 43. Méthode des TOPs k (Mesure Globale Sans Contexte)

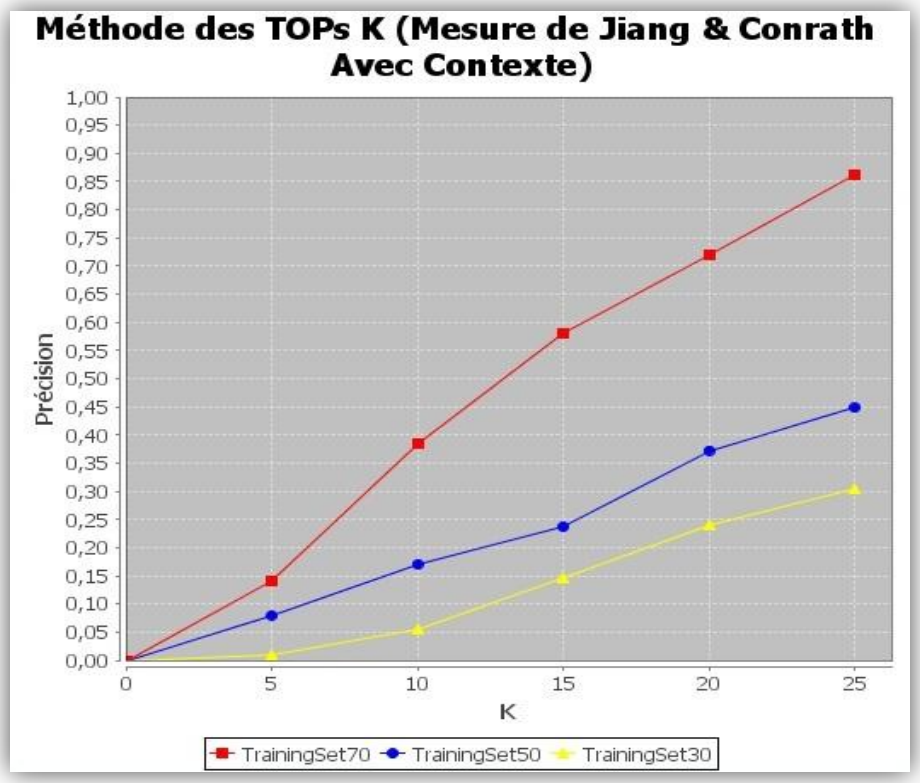

Figure 44. Méthode des TOPs k (Mesure de Jiang & Conrath Avec Contexte)

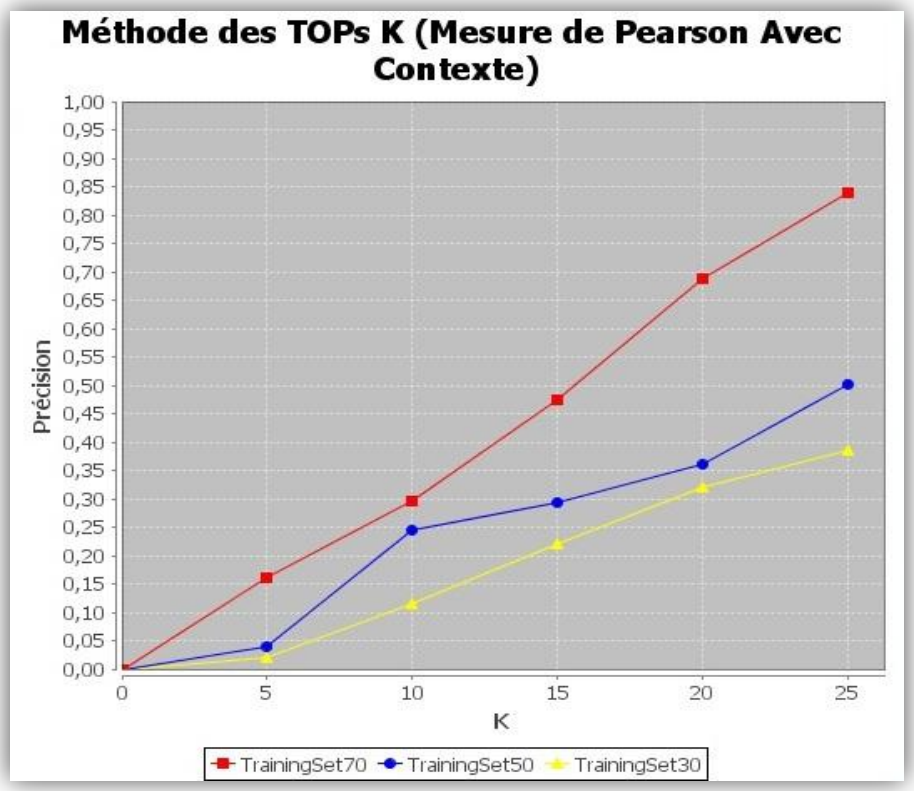

Figure 45. Méthode des TOPs k (Mesure de Pearson Avec Contexte)

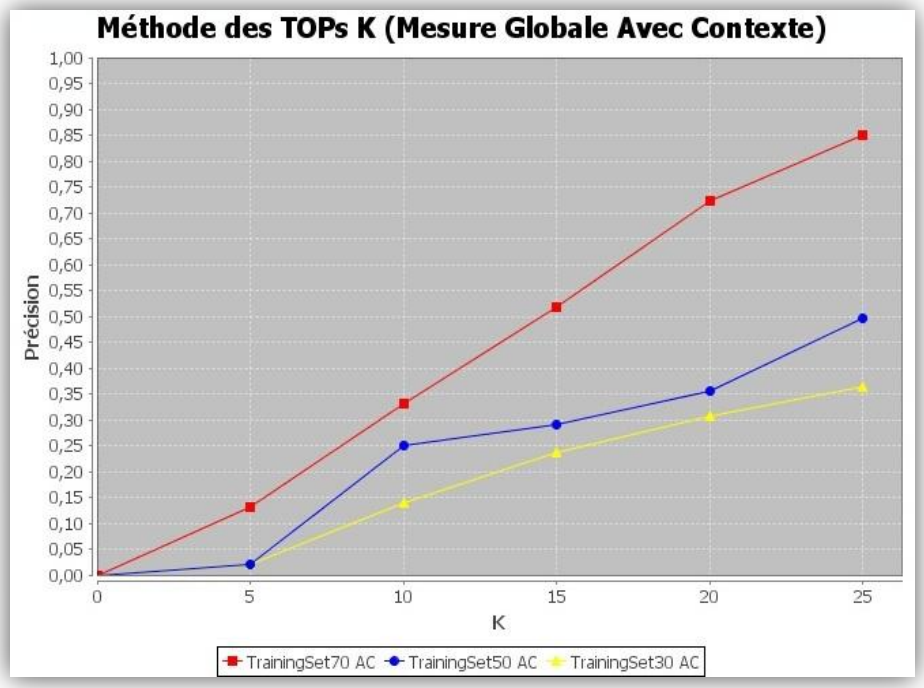

Figure 46. Méthode des TOPs k (Mesure globale Avec Contexte)

Afin de comparer la méthode contextuelle et la méthode non contextuelle, nous avons pris la répartition (70 :30), 70 films pour la construction des profils (Training_set) et 30 films pour le test (Result_set). Les figures 47, 48 et 49 montrent les courbes de Training_set70 en utilisant les trois types de similarité citées dans la section 3.2.

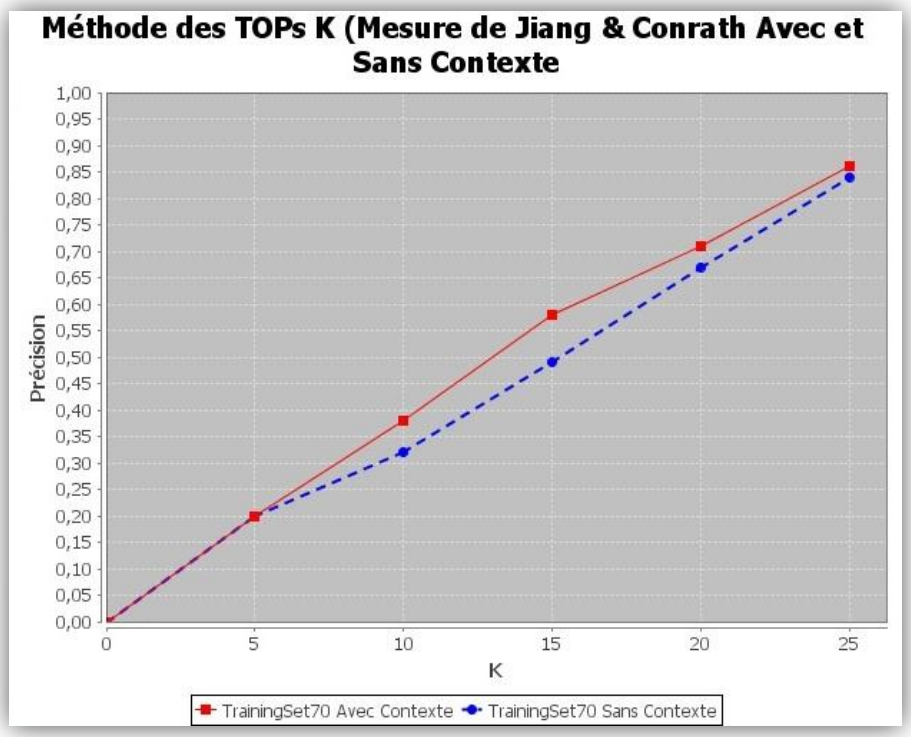

Figure 47. Méthode des TOPs k (Mesure de Jiang & Conrath Avec et Sans Contexte pour TrainingeSet 70)

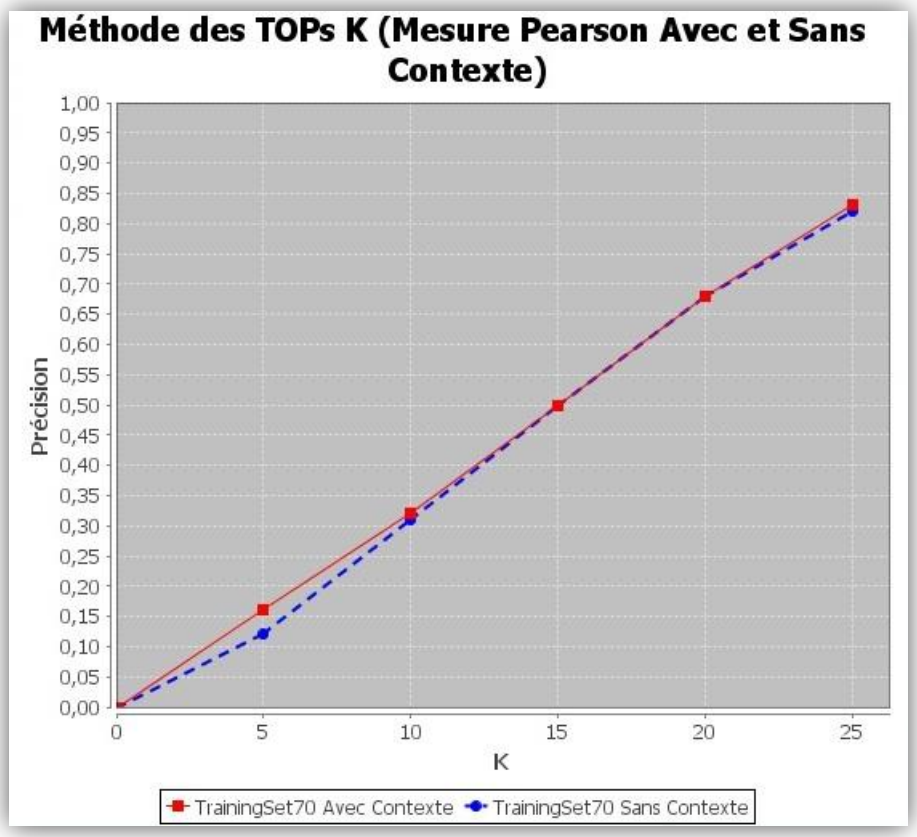

Figure 48. Méthode des TOPs k (Mesure de Pearson Avec et Sans Contexte pour TrainingSet 70)

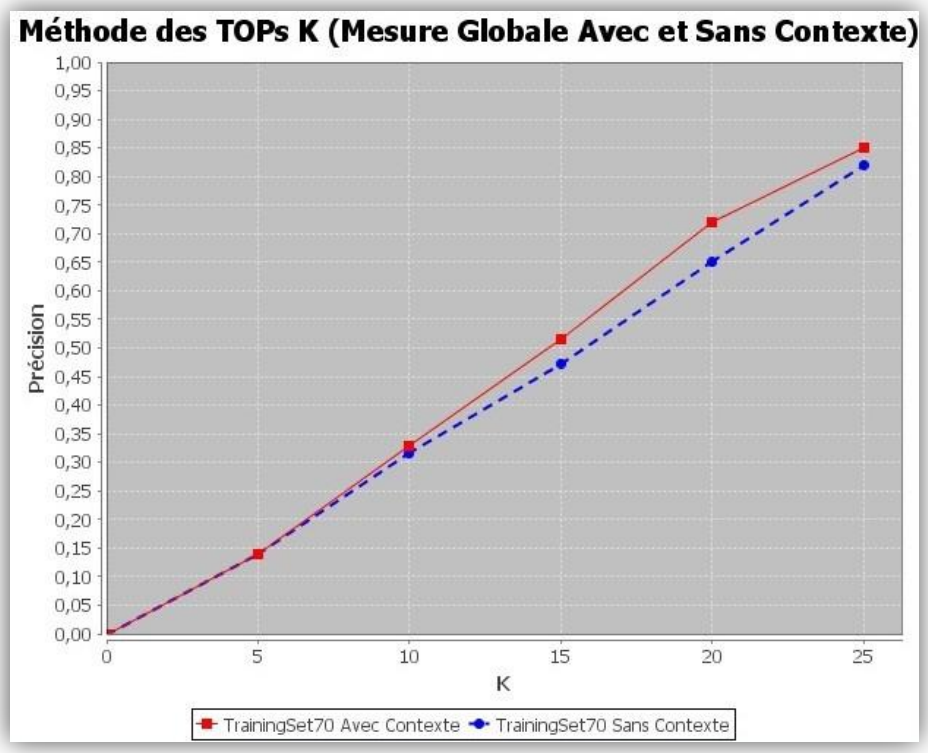

Figure 49. Méthode des TOPs k (Mesure Globale Avec et Sans Contexte pour TrainingeSet 70)

Nous remarquons d'après ces courbes, que la précision augmente en fonction de la taille de la table Training_set. Plus la taille Training_set augmente plus la précision est meilleure. En effet, la table Profil_user dépend de la taille de la table Training_set. Le profil d'un utilisateur est d'autant plus fourni qu'on a d'informations dans la table Training_set. Nous remarquons également que la méthode contextuelle fournit de meilleurs résultats par rapport à la méthode non contextuelle pour le cas Training_set70. Nous pouvons en déduire que l'utilisation du contexte à un impact positif sur la recommandation.

3.5.2 Résultats de la méthode du seuil

Pour la méthode du seuil, nous avons considéré les deux métriques : précision et rappel. Nous avons fait plusieurs expérimentations en effectuant des changements sur les tailles de la table Training_set pour un seuil fixé. Les tailles de la table Training_set retenues sont : 30, 40, 50, 60, 70, 80 et 90. Nous avons représenté les résultats d'expérimentation par des courbes. Chaque courbe nous donne la valeur de la précision ou le rappel en fonction de la variation de la taille de la table Training_set pour un seuil précis. La valeur du seuil considérée est : (seuil-note=3 ; seuil-prédiction=0,5), Les figures 50 et 51, présentent les courbes correspondant à la similarité globale avec coef1=0,75 et coef2= 0,25.

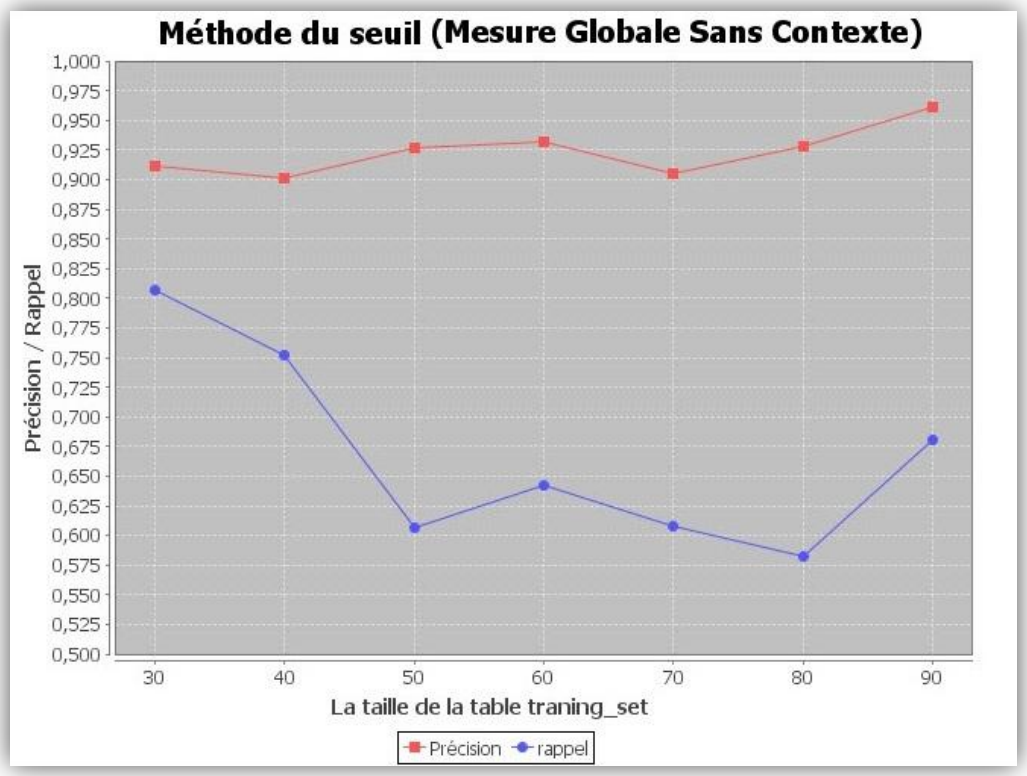

Figure 50. Méthode du seuil (Mesure Globale Sans Contexte)

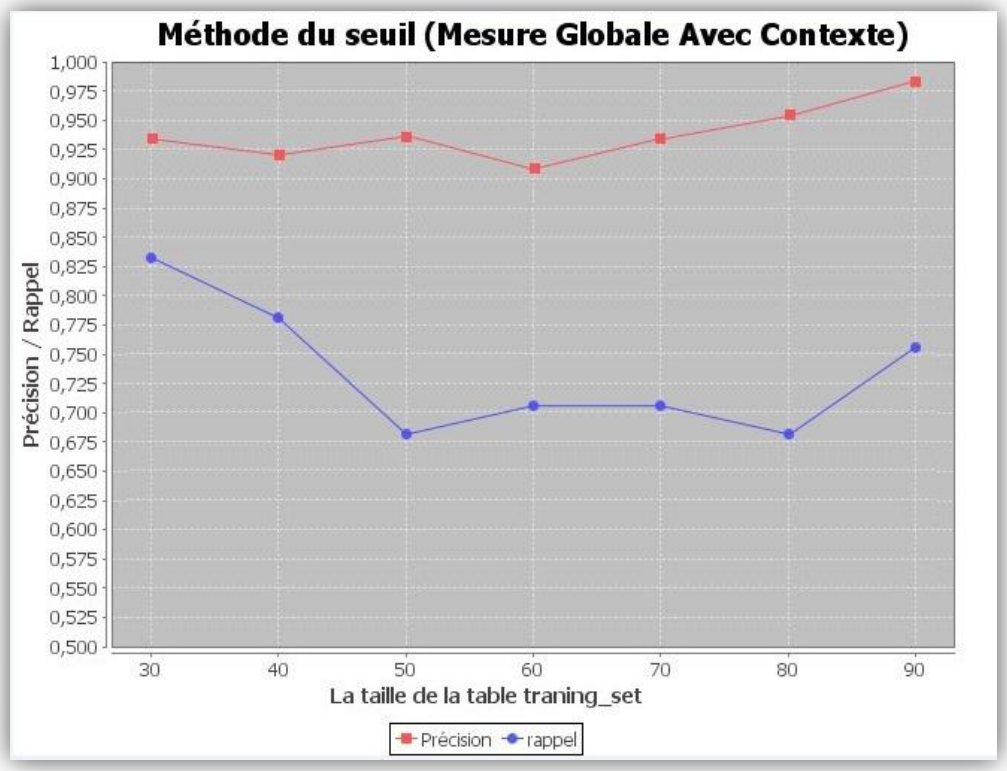

Figure 51. Mesure du seuil (Mesure Globale Avec Contexte)

D'après les figures 50 et 51, nous constatons que la précision et le rappel fourni par la mesure globale avec contexte sont meilleurs par rapport à ceux donnés par la mesure globale sans contexte.

4. Conclusion

Dans ce chapitre nous avons abordé l'évaluation du système de recommandation, suivant deux méthodes (seuil, top(k)), L'expérimentation nous a montré que la contextualisation a un impact positif sur le processus de recommandation.

CONCLUSION GENERALE

La personnalisation de l'information émerge comme une approche capitale dans le développement des systèmes du futur. Parmi ces derniers, nous pouvons citer les systèmes de recommandation qui donnent de l'importance aux préférences des utilisateurs, dans le but de leur proposer des ressources à acheter ou à consulter répondent au mieux à leurs besoins. Ils sont devenus des efficaces dans le commerce électronique, la recherche documentaire, le tourisme, etc., en fournissant des suggestions pertinentes au sein d'une grande masse d'informations.

L'intégration du contexte dans les systèmes de recommandation classiques, permet d'améliorer les performances de ces derniers. Nous avons présenté, dans ce mémoire, une nouvelle approche de la prise en compte de l'information contextuelle dans les systèmes de recommandation. Elle est générale dans le sens où elle peut être appliquée pour la recommandation de contenus en général (livre, Url, article, produit, film, chanson, etc.). Dans cette approche, le contexte est défini du point de vue des objets et non pas des utilisateurs. Il est de type intrinsèque, c'est-à-dire, défini sur l'ensemble des attributs relatifs à l'objet lui-même. Nous avons montré, en le combinant avec le profil de l'utilisateur, comment il peut améliorer la précision de la recommandation et ainsi mieux répondre aux exigences des utilisateurs.

L'architecture du système utilisant cette nouvelle méthode comporte trois composants essentiels, à savoir, *Création des profils utilisateurs*, *Résolution du contexte*, et l'*appariement profil/contenu*. Un premier algorithme a été réalisé dans le but de créer de façon automatique des profils utilisateurs sur la base de l'analyse des logs d'interaction. Un deuxième algorithme a été élaboré pour permettre l'extraction du contexte intrinsèque directement à partir des contenus identifiés comme candidats potentiels à la recommandation. Ce deuxième algorithme résout même le problème des contextes qui ne sont pas compatibles avec le profil de l'utilisateur, en se basant sur la similarité entre les instances du contexte. En ce qui concerne l'appariement profil/contenu, nous avons conçu et utilisé une nouvelle mesure de similarité qui combine linéairement une mesure sémantique et une mesure vectorielle.

Pour mieux comprendre les différents aspects pris en compte par notre approche, nous l'avons appliquée pour la recommandation de films à un utilisateur en choisissant comme contexte le réalisateur d'un film. Nous avons, également, présenté et comparé les principaux méthodes et systèmes sensibles au contexte sur la base d'un certain nombre de propriétés que nous avons définies.

Finalement, en vue de confirmer l'efficacité de notre application, des tests ont été effectués, sur les données récoltées de deux grandes bases de données des films à savoir MovieLens et IMDB, en utilisant l'approche de recommandation proposée. Ces tests ont montré l'importance du contexte dans un système de recommandation. L'évaluation du système avec contexte a révélé de meilleurs résultats en les comparant à ceux de l'évaluation d'un SR classique (sans contexte).

Ce travail est loin d'être terminé. Des perspectives sont à envisager, comme :

- Introduire le contexte dans les SR utilisant le filtrage collaboratif et le filtrage hybride,
- Intégrer le contexte extrinsèque qui est un ensemble d'attributs relatifs à l'environnement d'interaction de l'utilisateur dans le processus de recommandation.
- Combiner les deux types de contextes à savoir le contexte extrinsèque et le contexte intrinsèque afin d'affiner la liste des produits à recommander.

BIBLIOGRAPHIE

[1] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles. Towards a better understanding of context and context-awareness. In HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, pages 304-307, London, UK, 1999. Springer.

[2] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. ACM Trans. Information System, 23(1):103–145, 2005.

[3] G. Adomavicius, A. Tuzhilin. Context-Aware Recommender Systems, 2nd ACM International Conference on Recommender Systems, October, 2008.

[4] Amazon : www.amazon.com

[5] S.S. Anand, B. Mobasher. Contextual Recommendation. In Lecture Notes In Artificial Intelligence, volume 4737, pages 142-160. Springer-Verlag, Berlin, Heidelberg, 2007.

[6] L. Baba-Hamed, S. Abbar, R. Soltani et M. Bouzeghoub, "Elaboration et Evaluation d'un Système de Recommandation Sémantique", in proc. 1st international Conference on Information Systems and Technologies (ICIST'11), PP.515-523, 24-26 April, 2011.

[7] L. Baltrunas. Exploiting contextual information in recommender systems. In RecSys '08, pages 295–298, New York, NY, USA, 2008. ACM.

[8] L. Baltrunas, X. Amatriain, Towards Time-Dependant Recommendation based on Implicit Feedback, cars 2009.

[9] A. Benayache. Construction d'une mémoire organisationnelle de formation et évaluation dans un contexte e-learning : le projet MEMORAe. Thèse en Informatique de l'Université du Maine. 2005.

[10] S. Bouakaz. Système de recommandation sensible aux contextes. Mémoire de fin d'études. Septembre 2010.

[11] L. Candillier. Apprentissage automatique de profils de lecteurs. Mémoire de DEA. Laboratoire d'Informatique Fondamentale de Lille. Juin 2001.

[12] I. Cantador, P. Castells, Semantic Contextualisation in a News Recommender System. Decembre 2009.

[13] S. Castagnos. « Modélisation de comportements et apprentissage stochastique non supervisé de stratégies d'interactions sociales au sein de systèmes temps réel de recherche et d'accès à l'information », thèse de doctorat de l'université Nancy 2. Novembre 2008.

[14] S. Chatterjee, A. S. Hadi, and B. Price. Regression analysis by example. JohnWiley and Sons, 2000.

[15] M. Daoud, Recherche contextuelle d'information. RJCRI 2007.

[16] B. De Carolis, I. Mazzotta, N. Novielli and V. Silvestri, Using Common Sense in Providing Personalized Recommendations in the Tourism Domain. Septembre 2009.

[17] M. Domingues, A. Mário Jorge, C. Soares, Using Contextual Information as Virtual Items on Top-N Recommender Systems, cars 2009.

[18] F. Gandon. Ontologies informatiques. Thèse de doctorat. Mai 2006.

[19] N. Guarino. Formal Ontology and Information Systems. Conférence. Proceedings of FOIS'98. Pages 3-15. Trento, Italy. 6-8 Juin 1998.

[20] T. Hussein, T. Linder, W. Gaulke, J. Ziegler. Context-aware recommendations on rails. Septembre 2009.

[21] Inote: Image Annotation Tool. http://jefferson.village.edu/iath/inote.html.

[22] J. Jiang et D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In Proceedings of the 10th International Conference on Research in Computational Linguistics, Taiwan. 1998.

[23] D. Koller and M. Sahami. Toward optimal feature selection. In Proceedings of the 13th International Conference on Machine Learning, pages 284–292. Morgan Kaufmann, 1996.

[24] Limites des systèmes de recommandation : http://ziedzaier.com/category/systeme-de-recommandation.

[25] D. Lin. An Information-Theoretic Definition of similarity. In Proceedings of the Fifteenth International Conference on Machine Learning. Pages 296 - 304. 1998.

[26] H. Liu and H. Motoda. Feature selection for knowledge discovery and data mining. Springer, 1998.

[27] S. Lombardi, S. S. Anand, M. Gorgoglione. Context and Customer Behavior in Recommendation. Septembre 2009.

[28] Kr. Markov et Kr. Ivanova. An ontology- content-based filtering method Proceedings of the Fifth International Conference "Information Research and Applications", Varna, Bulgaria. Juin 2007.

[29] NetBeans : www.netbeans.org

[30] A. T. NGUYEN. COCoFil2 : Un nouveau système de filtrage collaboratif basé sur le modèle des espaces de communautés. Thèse. Université joseph Fourier – Grenoble I. Novembre 2006.

[31] Oracle: www.oracle.com

[32] U. Panniello, A. Tuzhilin, M. Gorgoglione, Experimental Comparison of Pre- vs. Post-filtering Approaches in Context-Aware Recommender Systems. 2009

[33] V. Peralta."Extraction and Integration of MovieLens and IMDb Data", technical report, APMD project , Laboratoire PRiSM, Université de Versailles .July 2007.

[34] P. Resnik. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. Journal of Articial Intelligence Research 11. Pages 95-130. 1999.

[35] T. Slimani, B. Ben Yaghlane et K. Mellouli. Une extension de mesure de similarité entre les concepts d'une ontologie. 4rth International Conference: Sciences of Electronic, Technologies of Information and Telecommunications. Mars 2007.

[36] Sourcetone : www.sourcetone.com

[37] L. Baba-Hamed, R. Soltani et Sabri K. Construction d'une ontologie pour la recommandation de films à un utilisateur. Actes des Ateliers des 21es Journées Francophones d'Ingénierie des Connaissances (IC 2010), Nîmes, France, juin 2010.

[38] R.Soltani, L.Baba Hamed, et S.Abbar. Contextualisation des préférences pour les systèmes de recommandation, 2nd International Conference on Information Systems and Technologies (ICIST'12), Sousse, Tunisia, March 24-26 2012 (www.icist2012.org). ISBN: 978-9938-9511-2-7.

[39] Z. Wu et M. Palmer. Verb semantics and lexical selection. Conference. In Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics. Pages 133-138. 1994.

ANNEXE 1

Manuel d'utilisation de l'application

Dans cette section nous montrons comment utiliser notre application.

Pour commencer nous devons choisir le type de session : Administrateur ou Utilisateur (voir figure 52).

Figure 52. Choix d'une session

Pour le choix de la session « Administrateur », la fenêtre de la figure 53 s'affiche pour l'authentification. Il faut donc entrer le nom d'utilisateur « Admin » ainsi que le mot de passe.

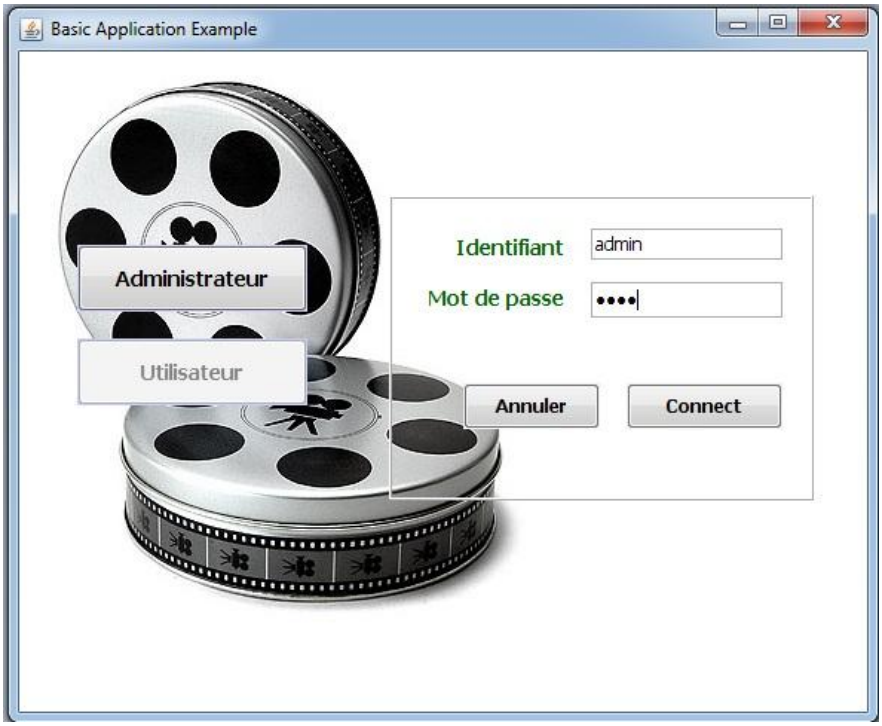

Figure 53. Authentification de l'administrateur

Après validation, une fenêtre s'affiche pour le choix de la méthode de prédiction (avec contexte ou sans contexte) comme il est montré dans la figure 54.
L'administrateur à la possibilité aussi de visualiser la matrice des réalisateurs, la recherche d'un profil utilisateur, et d'appliquer la prédiction globale ou partielle.

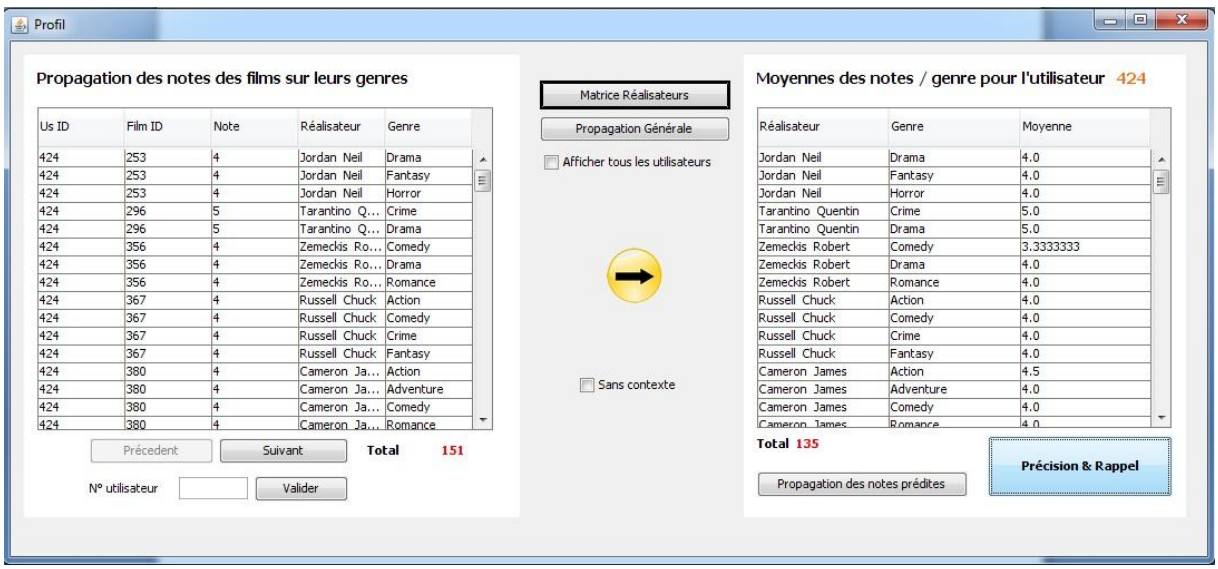

Figure 54. Choix de la méthode de prédiction

Après introduction du choix du matching, il faut sélectionner la méthode d'évaluation souhaitée (Méthode des TOPs K, Méthode du SEUIL), comme montré dans la figure 55.

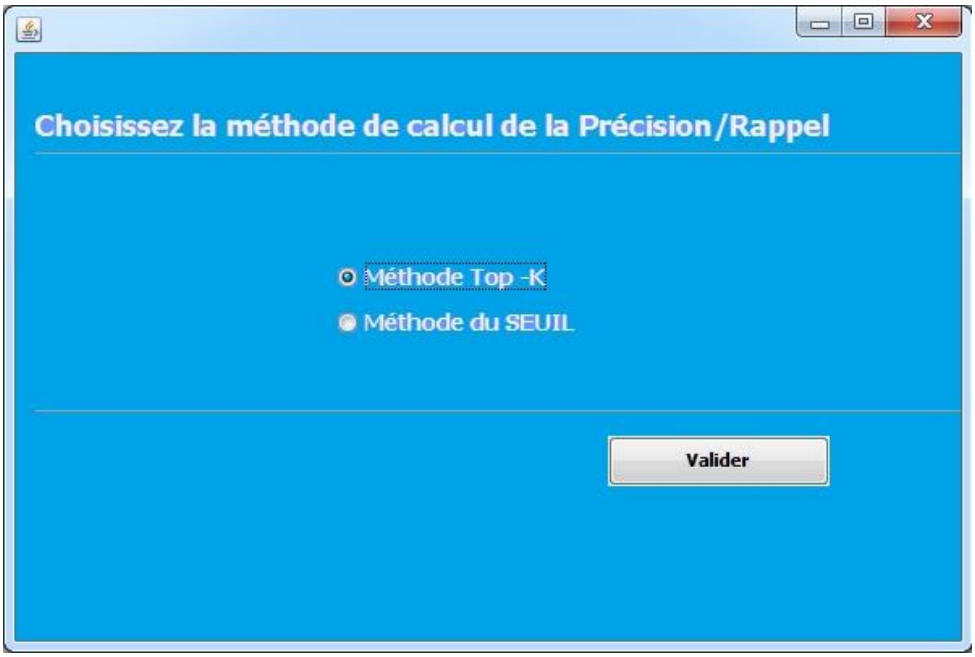

Figure 55. Choix de la méthode de prédiction

Si l'administrateur choisit la méthode des TOPs K, alors il a la possibilité d'introduire le nombre de k et les numéros de Training_Set souhaités (voir figure 56). Après avoir cliquer sur le bouton « Calculer », l'administrateur a la possibilité de visualiser les courbes correspondantes en cliquant sur le bouton « Graphe »

Figure 56. Méthode des TOPs

Dans le cas où l'administrateur choisit la méthode de Seuil, alors il a la possibilité d'introduire le seuil des notes données et le seuil des notes prédites comme montré dans la figure 57.

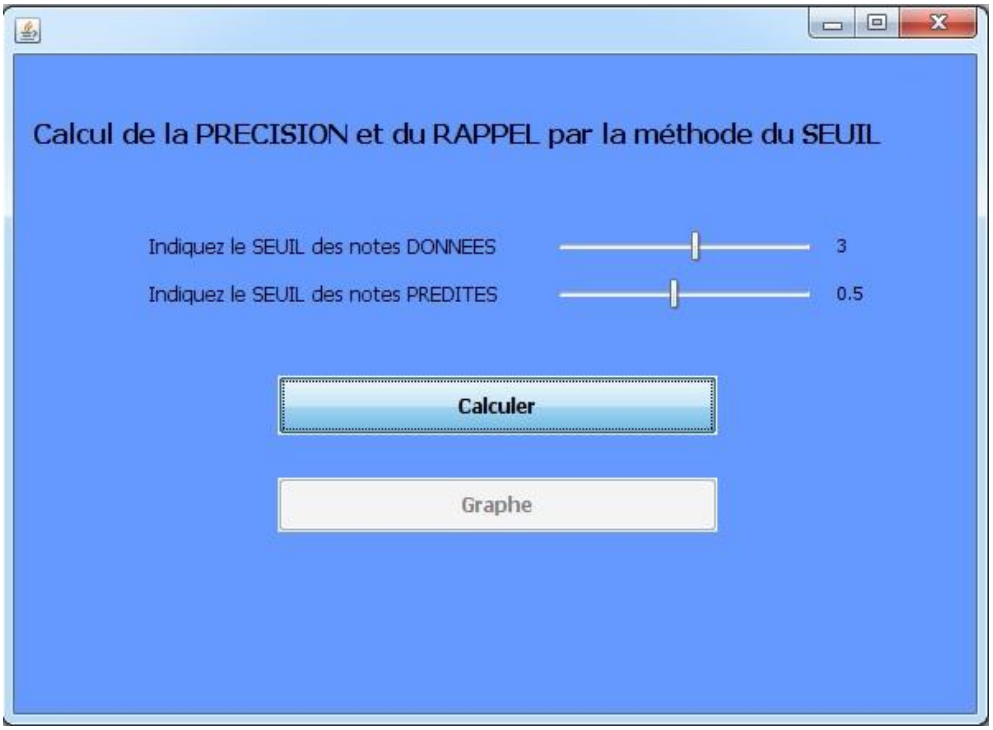
Calcul de la PRECISION et du RAPPEL par la méthode du SEUIL
Indiquez le SEUIL des notes DONNEES
3
Indiquez le SEUIL des notes PREDITES
0.5
Calculer
Graphe

Figure 57. Méthode de Seuil

Pour le choix de la session « Utilisateur », la fenêtre de la figure 58 s'affiche pour l'authentification. Il faut donc entrer l'identifiant de l'utilisateur ainsi que le mot de passe correspondant.

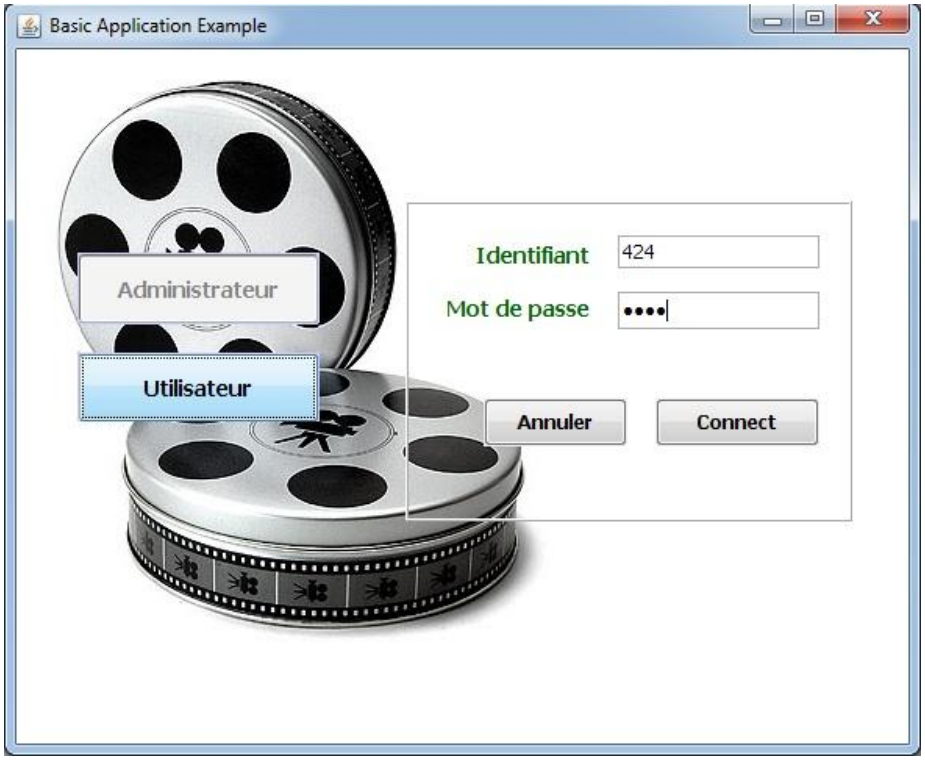

Figure 58. Authentification de l'utilisateur

Après l'authentification de l'utilisateur, une liste de films pertinents est affichée avec leurs détails (figure 59).

Figure 59. Liste des films pertinents

ANNEXE 2

Nous présentons dans ce qui suit un tableau qui contient les 100 films constituant notre ontologie avec leurs identifiants, ainsi que leurs genres.

Tableau 26. Représentation des films avec leurs genres respectifs

ID_Film	Titre	Genre
110	Braveheart	'Action', 'Biography', 'Drama', 'History', 'War'
253	Interview with the Vampire	'Drama', 'Fantasy', 'Horror'
260	Star Wars: Episode IV - A New Hope	'Action', 'Adventure', 'Fantasy', 'Sci-Fi'
296	Pulp Fiction	'Crime', 'Drama'
349	Clear and Present Danger	'Action', 'Drama', 'Thriller'
356	Forrest Gump	'Comedy', 'Drama', 'Romance'
367	Mask	'Action', 'Comedy', 'Crime', 'Fantasy'
377	Speed	'Action', , 'Thriller'
380	True Lies	'Action', 'Adventure', 'Comedy', 'Romance', 'Thriller'
457	Fugitive	'Action', 'Crime', 'Drama', 'Thriller'
480	Jurassic Park	'Action', 'Adventure', 'Horror', 'Sci-Fi', 'Thriller'
589	Terminator 2: Judgment Day	'Action', 'Adventure', 'Sci-Fi', 'Thriller'
590	Dances with Wolves	'Adventure','Drama', 'Western'
592	Batman	'Action', 'Crime', 'Fantasy', 'Thriller'
593	Silence of the Lambs	'Crime', 'Thriller'
608	Fargo	'Crime','Drama','Thriller'
648	Mission: Impossible	'Action', 'Adventure', 'Mystery', 'Thriller'
733	Rock	'Action', 'Adventure', 'Thriller'
780	Independence Day	'Action', 'Sci-Fi', 'Thriller'
858	Godfather	'Crime','Drama'
919	Wizard of Oz	'Adventure','Family','Fantasy','Musical','Drama', 'Mystery'
923	Citizen Kane	'Drama', 'Mystery'
1036	Die Hard	'Action', ' Crime ', 'Thriller'
1080	Monty Pythons Life of Brian	'Comedy'
1097	E.T. the Extra-Terrestrial	'Drama','Family','Fantasy', 'Sci-Fi'
1127	Abyss	'Action', 'Adventure','Drama', 'Sci-Fi', 'Thriller'
1136	Monty Python and the Holy Grail	'Adventure', 'Comedy'
1193	One Flew Over the Cuckoos Nest	'Drama'
1196	Star Wars: Episode V - The Empire Strikes Back	'Action', 'Adventure',' Fantasy ', 'Sci-Fi'
1198	Raiders of the Lost Ark	'Action', 'Adventure', 'Thriller'
1200	Aliens	'Action', 'Horror', 'Sci-Fi', 'Thriller'
1208	Apocalypse Now	'Action', 'Adventure','Drama', 'War'
1210	Star Wars: Episode VI - Return of the Jedi	'Action', 'Adventure',' Fantasy ', 'Sci-Fi'
1213	GoodFellas	'Biography', 'Crime','Drama'
1214	Alien	'Horror', 'Sci-Fi', 'Thriller'
1219	Psycho	'Horror', 'Thriller'
1221	Godfather: Part II	'Crime','Drama'
1240	Terminator	'Action', 'Sci-Fi', 'Thriller'
1247	Graduate	'Comedy','Drama', 'Romance'
1252	Chinatown	'Crime', 'Mystery', 'Thriller'

1258	Shining	'Horror', 'Thriller'
1259	Stand by Me	'Adventure', 'Drama'
1270	Back to the Future	'Action', 'Adventure', 'Comedy', 'Sci-Fi'
1291	Indiana Jones and the Last Crusade	'Action', 'Adventure',' Fantasy ', 'Thriller'
1294	M*A*S*H	'Comedy','Drama', 'War'
1304	Butch Cassidy and the Sundance Kid	'Adventure','Biography','Crime', 'Western'
1371	Star Trek: The Motion Picture	'Adventure', 'Sci-Fi'
1374	Star Trek: The Wrath of Khan	'Adventure', 'Sci-Fi'
1380	Grease	'Comedy', 'Musical', 'Romance'
1387	Jaws	'Adventure', 'Horror', 'Thriller'
1394	Raising Arizona	'Action', 'Comedy', 'Crime'
1396	Sneakers	'Action','Adventure','Comedy','Crime','Drama','Mystery', 'Thriller'
1408	Last of the Mohicans	'Adventure', 'Romance', 'War'
1527	Fifth Element	'Action', ' Adventure ', 'Sci-Fi', 'Thriller'
1573	Face/Off	'Action','Crime','Drama', 'Sci-Fi', 'Thriller'
1580	Men in Black	'Action', 'Comedy', 'Sci-Fi'
1597	Conspiracy Theory	'Drama','Mystery','Romance','Thriller'
1610	Hunt for Red October	'Action','Adventure','Thriller'
1617	L.A. Confidential	'Crime','Drama','Mystery', 'Thriller'
1674	Witness	'Drama','Romance','Thriller'
1954	Rocky	'Action','Drama','Romance','Sport'
1958	Terms of Endearment	'Comedy','Drama','Romance'
1994	Poltergeist	'Horror', 'Thriller'
1997	Exorcist	'Drama','Horror', 'Thriller'
2000	Lethal Weapon	'Action', 'Comedy','Crime','Drama','Thriller'
2012	Back to the Future Part III	'Action', 'Adventure', 'Comedy', 'Sci-Fi', 'Western'
2054	Honey	'Adventure', 'Comedy', 'Family', 'Sci-Fi'
2088	Popeye	'Adventure', 'Comedy', 'Musical'
2115	Indiana Jones and the Temple of Doom	'Action', 'Adventure', 'Comedy', 'Thriller'
2174	Beetlejuice	'Comedy',' Fantasy', 'Horror'
2193	Willow	'Action', 'Adventure', 'Comedy',' Fantasy'
2194	Untouchables	'Crime','Drama','History', 'Thriller'
2405	Jewel of the Nile	'Action', 'Adventure', 'Comedy','Romance'
2406	Romancing the Stone	'Action', 'Adventure', 'Comedy','Romance'
2409	Rocky II	'Action','Drama','Sport'
2410	Rocky III	'Action','Drama','Sport'
2411	Rocky IV	'Action','Drama','Sport'
2470	Crocodile Dundee	'Adventure', 'Comedy'
2571	Matrix	'Action','Sci-Fi', 'Thriller'
2640	Superman	'Action', 'Adventure',' Fantasy ','Sci-Fi'
2641	Superman II	'Action', 'Adventure',' Fantasy ','Sci-Fi'
2710	Blair Witch Project	'Horror', 'Mystery', 'Thriller'
2716	Ghostbusters	'Animation', 'Comedy',' Fantasy ', 'Horror', 'Sci-Fi'
2746	Little Shop of Horrors	'Comedy', 'Horror','Musical','Sci-Fi'

2762	Sixth Sense	'Drama','Mystery', 'Thriller'
2797	Big	'Comedy', 'Drama','Family',' Fantasy ','Romance'
2871	Deliverance	'Adventure','Drama','Thriller'
2915	Risky Business	'Comedy','Crime'
2916	Total Recall	'Action', 'Adventure','Horror ','Sci-Fi', 'Thriller'
2918	Ferris Buellers Day Off	'Comedy'
2968	Time Bandits	'Adventure','Comedy','Family',' Fantasy '
2987	Who Framed Roger Rabbit?	'Animation', 'Comedy','Crime','Family',' Fantasy ','Mystery'
3039	Trading Places	'Comedy'
3256	Patriot Games	'Action','Crime','Drama','Thriller'
3421	Animal House	'Comedy'
3448	Good MorningVietnam	'Comedy','Drama','War'
3471	Close Encounters of the Third Kind	'Adventure', 'Drama','Sci-Fi'
3527	Predator	'Action','Horror ','Sci-Fi', 'Thriller'
3552	Caddyshack	'Comedy','Sport'